

RISK-BASED APPROACH TO ASSESSMENT OF ADVANCED
TECHNOLOGIES FOR CONCEPTUAL DESIGN

A Thesis

presented to

the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

by

Adipratnia Asmady

August 2016

© 2016
Adipratnia Asmady
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Risk-based Approach to Assessment of Advanced
Technologies for Conceptual Design

AUTHOR: Adipratnia Asmady

DATE SUBMITTED: August 2016

COMMITTEE CHAIR: Aaron Drake, Ph.D.
Associate Professor of Aerospace Engineering

COMMITTEE MEMBER: Dianne DeTurris, Ph.D.
Professor of Aerospace Engineering

COMMITTEE MEMBER: Robert McDonald, Ph.D.
Professor of Aerospace Engineering

COMMITTEE MEMBER: Kurt Colvin, Ph.D.
Professor of Industrial & Manufacturing Engineering

ABSTRACT

Risk-based Approach to Assessment of Advanced Technologies for Conceptual Design

Adipratnia Asmady

The conceptual design phase of an aerospace system development program is typically characterized by short duration and relatively limited resources, yet design decisions are made that have critical implications on program risk. To address the more aggressive requirements, one of these decisions is the selection of advanced technologies. System developers need to assess advanced technologies early on, but are faced with uncertainties surrounding the potential net benefits. The concept introduced in this study is uncertainty characterization as a way to better understand the associated risk. A framework was developed to guide the interaction between the technology developer and the system developer. The objective is to gain a more comprehensive landscape of the technology options by explicitly considering the effects of uncertainty in the decision making process. This can ultimately facilitate prioritization and resource management during conceptual design. An example case of advanced wing technology was applied to the design of a high-altitude long-endurance unmanned aerial vehicle to demonstrate the implementation of the framework.

Keywords: Conceptual design, advanced technologies assessment, systems engineering, uncertainty characterization, program risk management, complex systems design

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER	
1 CONCEPTUAL DESIGN OF COMPLEX SYSTEMS	1
1.1 Complex Systems	1
1.1.1 Complex Systems Theory	2
1.1.1.1 Nonlinearity	2
1.1.1.2 Diversity	3
1.1.1.3 Degree of Autonomy	3
1.1.1.4 Self-organization	4
1.1.2 Design of Complex Systems	4
1.1.2.1 Synergy and System Interaction	5
1.2 Conceptual Design	7
1.2.1 Role of Advanced Technologies	9
1.3 Research Objective	10
2 UNCERTAINTY MANAGEMENT FOR ADVANCED TECHNOLOGIES .	13
2.1 Uncertainty as Program Risk	13
2.2 Technology Assessment Techniques	19
2.3 Approach	20
3 EXAMPLE CASE OVERVIEW	23
3.1 System	23
3.2 Advanced Technology	27
4 DECISION MAKING FRAMEWORK	30
4.1 Design Objectives	32

4.1.1	Example Case	33
4.1.2	Key Idea and Applicability	34
4.2	Technology Vision	35
4.2.1	Example Case	35
4.2.2	Key Idea and Applicability	37
4.3	Applicability Assessment	38
4.3.1	Example Case	39
4.3.2	Key Idea and Applicability	39
4.4	Integration Requirements	41
4.4.1	Example Case	41
4.4.2	Key Idea and Applicability	43
4.5	Technology Roadmap	44
4.5.1	Example Case	44
4.5.2	Key Idea and Applicability	47
4.6	Technology Development Plan	48
4.6.1	Example Case	48
4.6.2	Key Idea and Applicability	50
4.7	Technology Interaction Map	51
4.7.1	Example Case	56
4.7.2	Key Idea and Applicability	60
4.8	Uncertainty Characterization	61
4.8.1	Example Case	64
4.8.2	Key Idea and Applicability	66
4.9	Program Risk Categorization	68
4.9.1	Example Case	71
4.9.2	Key Idea and Applicability	77
4.10	Decision Making Strategy	79
4.10.1	Acceptance	79
4.10.2	Avoidance	79
4.10.3	Protection	80
4.10.4	Reduction	80
4.10.5	Research	80

4.10.6 Reserves	80
4.10.7 Transfer	81
5 CONCLUSION	82
BIBLIOGRAPHY	84

LIST OF TABLES

Table		Page
3.1	Initial assumptions	26
3.2	Initial aircraft sizing	26
3.3	Characteristics comparison of elliptical and bell spanload [16].	29
4.1	Uncertainty Characterization	77

LIST OF FIGURES

Figure	Page
1.1 Typical timeline for aerospace programs.	7
2.1 State of information, adapted from Nikolaidis [9].	14
2.2 Program risk is more than maturity.	15
2.3 Potential implication of uncertainty as performance risk.	17
2.4 Elements of program risk.	18
2.5 Risk management process [14].	21
3.1 Program timeline.	25
3.2 Prandtl's elliptical and bell spanload explained [16].	27
3.3 Bell spanload wingtip characteristics. (a) Spanload downwash profile (b) Vortex rollup behind the wing, analytical and in flight (c) Formation flight of Brown Pelicans demonstrating wingtip overlap (d) Spedding's Kestrel data showing inboard vortex core location[16].	28
4.1 Technology assessment framework.	31
4.2 Example integration boundary for the advanced wing technology. . . .	42
4.3 Example timeline for HALE UAV program.	46
4.4 Example technology roadmap.	46
4.5 Identifying the technology's sphere of influence.	51
4.6 Technology map.	52

4.7	Subsystem interaction.	53
4.8	Physical component identification.	54
4.9	System performance metric identification.	54
4.10	High-level interaction.	56
4.11	Subsystem interaction.	57
4.12	Component interaction.	58
4.13	System interaction map.	59
4.14	Objective of uncertainty characterization.	61
4.15	Technology interaction map with uncertainty.	63
4.16	Uncertainty identification for system performance metric.	63
4.17	Uncertainty sources for Prandtl-Bowers wing example case.	64
4.18	Objective of requirements identification.	68
4.19	Assessing technology's degree of influence.	69
4.20	Requirement identification for each system performance metric.	69
4.21	Uncertainty characterization link.	70
4.22	Technology map with requirements for Prandtl-Bowers example.	72
4.23	Example of influential uncertainty.	73
4.24	Assessment of a technology-to-system link.	74
4.25	Example of benign uncertainty.	75
4.26	Assessment of a technology-to-system link.	75

Chapter 1

CONCEPTUAL DESIGN OF COMPLEX SYSTEMS

Engineering design is a process that involves an unstructured space of solutions and limited knowledge. System designers need to come up with solutions based on a series of decisions that considers the trade-off between opportunity and risk. Challenges emerge when the system being designed becomes more complex, and the risk and opportunity become more abstract. During conceptual design, the system developers need to deal with the complexity when decisions need to be made with the least available knowledge. While there is high degree of design freedom, there is high risk associated with the decisions that can have a significant impact on the subsequent phases of the program.

1.1 Complex Systems

Aerospace systems are a representative example of complex systems. The design of aerospace systems is becoming more challenging as systems are expected to have more capabilities and higher autonomy. To illustrate the growth of complexity in systems being developed today, the fifth generation combat aircraft, such as the F-35, has 130 subsystems, on the order of 10^5 interfaces, and 90% of its functions managed by software. This is a substantial growth from the fourth generation aircraft, such as the F-16, that has 15 subsystems, 10^3 interfaces, and 40% of its functions managed by software [1]. One magnitude growth in subsystem resulted in an increase in interfaces by two magnitudes, meaning there are about a hundred times more sources of potentially unpredictable interaction.

During the design phase, the system developers do not deal with the complexity of the physical system, but more in the complexity of the decision making process. Complex systems design involves having to consider multiple factors concurrently and the implicated risks. At the conceptual design stage, the risk comes from having to make decisions when there is minimal knowledge. The system developer needs to determine a design strategy that ultimately shapes how the program materializes. The implication of the decisions made during conceptual design can have an enduring impact in the program, yet these decisions are made when the uncertainty is at its greatest.

As a way to set the backdrop of this study, concepts from complex system theory are presented here to introduce the challenges in the design of complex systems. This preliminary discussion unpacks the definition of complex system to draw some connection between the features of complex system and how they embody uncertainty.

1.1.1 Complex Systems Theory

A system can simply be defined as a set of things that are arranged in a particular combination to serve some functionalities [2]. With elements that comprise a set of properties, simple systems are closed, uniform, and predictable. The fundamental idea is that a system is the sum of its constituents and can be designed in a top-down manner. A complex system, on the other hand, is a system that is said to have an emergent behavior. Emergent behavior refers to a property of the system that can only be studied at a higher level, and relates to a function that none of its individual constituents possess. The features of a complex system that lead to the rise of new global behavior are nonlinearity, diversity, autonomy, and self-organization [3].

1.1.1.1 Nonlinearity

One of the biggest challenges in trying to understand complex systems is the existence of nonlinear relationships. In a linear-thinking mind, if a small push produces small response, it is intuitive to think that pushing twice as hard will produce a response that is twice as big. Yet, since the world is full of nonlinearities, pushing twice as hard can lead to a response that is three or four times as big [2]. It is difficult to draw

cause-effect relationships between two components, and if the system has thousands of parts, then the nonlinearities become even less manageable. Nonlinearity can also flip a system from one mode to another. This gives rise to uncertainties in the system-level behavior when the characteristics of the interdependence, and their limits, are not fully known.

1.1.1.2 Diversity

Another property of a complex system is diverse components. The system elements typically have a unique set of properties that are combined into a heterogeneous organization [3]. The number of interactions between system elements can grow exponentially as the number of elements increase, and it easily becomes strenuous to keep track of all the interactions. Diverse components produce uncertainties in what relationships actually exist within the system. While theoretically the properties of the system element can be known exclusively, it is not always certain that the properties of the element will remain the same when it is combined with other elements. System interactions can lead to unexpected behavior simply due to unknown relationship between the properties of two or more elements.

1.1.1.3 Degree of Autonomy

Autonomy is another feature that contributes to complexity. When the system is designed to have a low level of autonomy, the system can be design, managed, and controlled in a top-down fashion. However, as autonomy is increased, this top-down approach becomes less practical because the control and organization become distributed [3]. It is ultimately the increasing interactions at the local level that come to define how the system develops. Here, uncertainty is present in terms of how the input and output of the system will be processed. With a low level of autonomy, an input can be designed to only yield a single output. Alternatively, a high degree of autonomy transfers the decision making capability to the system elements. This means that the element can process different inputs and produce different outputs. While rigid protocols can be made, uncertainties in how they will be executed still exists. A small mishap in the execution can lead to erratic behavior in the system.

1.1.1.4 Self-organization

When the system elements have the autonomy to adapt locally, they can self-organize to form global patterns. This process is referred to as emergence. As opposed to linear systems, where order usually comes from a top-down manner, the patterns of order for complex systems emerge from the bottom up [3]. Combining the previous three characteristics, self-organization depends on the properties of the elements and their interactions. If the relationship between system elements is not fully understood, it produces uncertainties in the type of emergent properties that will be observed. The elements will simply respond to its immediate surroundings within the system architecture, and the resulting local behaviors can interact to give rise to the global response that is different than expected.

1.1.2 Design of Complex Systems

From looking at the properties of complex system, it can be observed that complexity partly manifests itself in terms of uncertainties in the system behavior. Complex systems are deeply influenced by the interactions of the system elements, hence optimum systems do not come from individually optimized components. This means that the system designer needs to configure the context and the local interactions that can lead to effective global coordination. The system developer has to recognize the two qualitatively different levels of a system, and design to multi-dimensionality.

Traditional paradigm uses the reductionist approach, where systems are designed using a top-down plan, and the system is decomposed into elements with independent sub-functions. This works well when the elements are stable, predictable, and controllable. For complex systems, however, reductionism does not work well as well because there is greater emphasis on the interconnectedness and degree of autonomy. Complex systems are typically open systems, and they have dynamic networks with boundaries that are not well-defined [3].

Because complex systems cannot be understood at a single scale of analysis, systems designers must develop the habit of looking at their system at many different scales, by iteratively zooming in and zooming out [4]. The system developer needs

to ask whether the problems can be solved more effectively by addressing them at a higher or lower hierarchical level. The systems designers must be especially open to solutions that arise from the bottom-up through self-organization, rather than only seeking to impose order from the top-down.

This is where the challenges in designing complex system emerge. The system developer needs to manage two different levels of behavior, while having to deal with uncertainties in the interactions and properties of the elements. Additionally, the design of complex systems is driven by functionality; the customer or end-users are less concerned about the actual system and its constituents, but are more interested in whether or not the system delivers the desired capability and its cost-effectiveness. As system designers, the focus should then be in trying to leverage the constructive relations of system elements by finding a balance between the component properties and the network that provides structure to the system.

1.1.2.1 Synergy and System Interaction

A design solution is considered successful if all the elements work in synergy to provide the desired function. In other words, synergy is what gives systems their emergent property. Synergy can be defined as the interaction or cooperation of two or more organization, substances, or other agents to produce a combined effect greater than the sum of their separate effects [3]. Consider a musician who combines three songs by three different artists. The musician needs to somehow overcome the difference, diversity, and general messiness of the three songs to produce something that sounds authentic and in harmony. Similarly, system developers need to combine different technologies that generally have different characteristics and purposes, and create a system that provides certain functionality. However, to produce an elegant design solution, the system developers need to think about how to leverage the synergy of the elements and the network structure.

Analogous to designing an aircraft for stability, designing for synergy means to design for a state that the elements of the system will naturally gravitate towards [3]. This requires investment on designing the relations and infrastructure at which the system interacts. At the early stage of the design, the specifics of the system

elements are still rudimentary, so it is difficult to know how the system elements will interact with one another, let alone whether or not things will exhibit synergistic behavior. Nonetheless, it is critical for the system designers to look at how the flow of information should be designed, and how changes in the states of the components can influence the network. More importantly, a lot of uncertainties reside in the interactions between elements and their interdependence. This puts a stronger emphasis in the need for the system developers to design from the lens of network infrastructure.

In a complex system, it matters less how good the properties of the component are, it is the properties of the network will allow the system to produce an emergent behavior. Consider for example, a sports car with high-end engine performance. It will perform great when it is driving on a highway, but when it is in urban traffic, it will go as fast as the traffic allows. This suggests that while the properties of the element are important, the structure and dynamics of the network are central to the solution.

From a system-thinking approach, it is important to see complex engineered systems from the perspective of connectivity. Irrespective of whether it is called network or system of systems, network is arguably the true geometry behind complex systems [3]. The design process should therefore be about designing for access depending on where the element lies in the network. This defines the flow of information that ultimately leads to the global behavior of the system.

1.2 Conceptual Design

Due to its complexity and relative length of aerospace programs, the development of aerospace systems are divided into different stages, typically consisting of design, testing, production, and operation (see Fig. 1.1). The design phase is usually further organized into three phases: conceptual, preliminary, and detailed design. This is done to allow the design work to be more focused and have progressive advancement towards increasing detail. The conceptual design phase is generally the shortest duration, while the detailed design phase is the longest. To put it in perspective, the conceptual design phase can take between 3 months to a year, whereas detailed design can last up to 3-5 years. In this study, the decision making situation of interest takes place in the conceptual design phase.

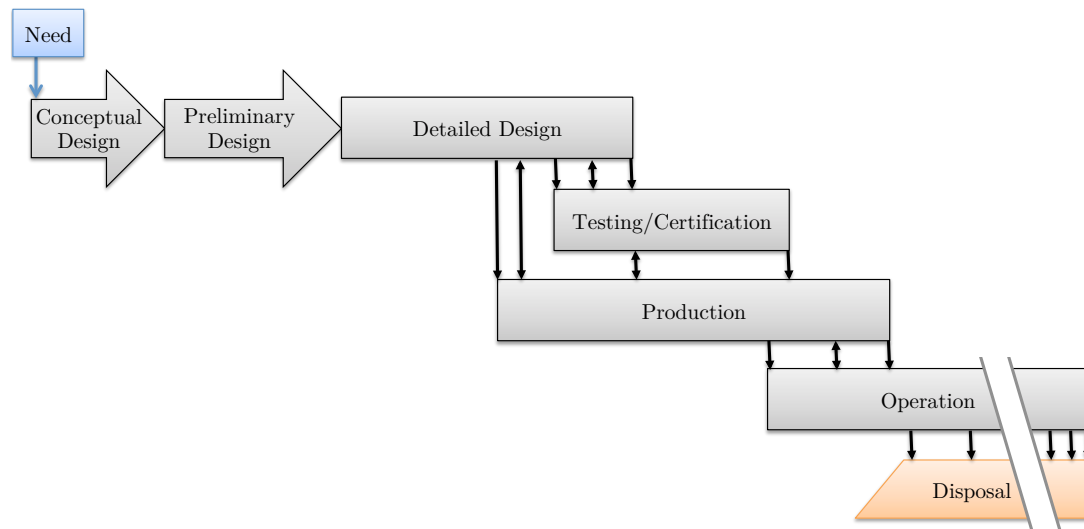


Figure 1.1: Typical timeline for aerospace programs.

The design process usually begins with functional requirements that are formed on the basis of developing a system that exceeds the performance of previous ones or serves a new function. The first question to ask is typically whether or not a new system is really required, or if an existing one can be used with slight modification. The conceptual design phase is initiated once the user-needs are clearly established. Each of the design phase is intended to address different objectives, hence producing different parts of the system definition at varying level of detail.

Some of the questions that need to be answered during conceptual design include:[5]

- Will the system work?
- What does the system look like?
- What requirements drive the design?
- What trade-offs should be considered?
- What should the system weigh and cost?

Conceptual design is a very fluid process. New ideas and problems emerge as a design is investigated in ever-increasing detail. The primary objective for conceptual design is to determine if there is a sound physical solution to the functional needs, in essence, answering a feasibility question [5]. The conceptual design phase is characterized by high degree of design freedom; it is arguably about formulating the ‘starting point’, from initial interpretation of the requirements to initial assumptions for guiding the analysis. The design team, however, needs to make decisions when there is the least available knowledge about the system, which to some extent will impact the subsequent phases. Consequently, there is high risk associated with decisions made during conceptual design. Due to the generally short duration of this phase, the system developer also does not have much time or resources to gather extensive information, which imposes additional challenge to the decision making process.

With crude information about the system, the system developer needs to deal with uncertainties that naturally introduce risk to the program. One of the systems engineering tasks during conceptual design is to develop program planning and devise a strategy for the design process. To do this effectively, the systems engineers need to explicitly acknowledge how decisions are bringing risk into the program. Many argue that this is where the most common seeds of troubled programs are sown [6]. If the system developer is not cautious when assessing program risk, the consequence of a misjudgement can manifest itself later resulting in delays, budget overruns, program cancellation, or a system that does not perform to requirements.

1.2.1 Role of Advanced Technologies

One of the decisions that need to be made during conceptual design is the selection of advanced technologies. To satisfy aggressive requirements, the system must incorporate new technologies, which is one of the motivations for advanced technologies research. Advanced technologies are fundamentally based on the idea that technological advances must be pursued and integrated to deliver a product that is superior relative to the current state-of-the-art. There are strong incentives to develop new technologies, such as increased capability or market share. However, new technologies deal with a variety of uncertainties, starting with the uncertainty in its net integrated benefits.

System developers naturally tend to adhere to incremental improvements rather than significant advances due to the risk and investment cost involved [7]. Depending on the system, advanced technologies can take on very different roles in the program. For large aerospace programs, which can take 5 to 15 years to develop, it is more critical to determine early on which advanced technologies to include, so that they can be developed in parallel with the system. Financial commitments are usually established during the earlier phase of the design, meaning that new technological solutions have to be identified and assessed when the business case is being determined.

Due to finite resources, design engineers cannot study each of the potential technologies extensively. Sufficient information, however, must be collected so that well-founded evaluation can be done. For conceptual design, this means that one of the most important tasks is to maximize the quality of the information being exchanged, and to transform it into a form that can be used to support decision making. Considerable uncertainties surround advanced technologies due to the absence or lack of previous historical data, hence it should be a key feature to represent in the decision making process. With limited knowledge about the nature of the technology, there is uncertainty in how it will interact with other elements and the resulting behavior at the system-level. As will be discussed in Chapter 2, this uncertainty signifies risk to the program and needs to be explicitly considered during the evaluation.

Advanced technologies assessment involves two groups that need to be considered, the system developer and technology developer. The socio-technical reality to

acknowledge when it comes to technology development is that often times there is a disconnect between the two groups due to their differing background and perspective on the effects of the technology. The focus of the technologist is to explore a physical phenomenon and create a widget to exploit its potentials. The system developer, on the other hand, focuses on developing a system that can satisfy the customer requirements, where the main concern lies in the net benefit and the associated risk. During the initial interaction, the technology developer is then more inclined to present findings that are on the optimistic side. There is also less incentive for the technology developer to explicitly express the uncertainties. The system developer needs to subsequently interpret the information to make an objective judgement on the implications of the technologies selection.

1.3 Research Objective

When systems are complex, their structures cannot be described at a single level, and multiscale analyses are necessary to understand complex systems. Their emergent behavior, derived from the relationships among their elements and with the environment, gives rise to observed patterns that may not be understood or predicted. Uncertainty is embodied in complex systems because the level of information needed to describe these interactions as a response function is practically unobtainable. At the conceptual design stage, the uncertainty is at its greatest, thus system developers should try to manage it as they would manage risk. Inherently, risk cannot be eliminated, but can be managed and mitigated. All uncertainties have the potential to manifest themselves as a shortcoming. The assessment of advanced technology is used as a specific decision making case to illustrate the underlying philosophy of uncertainty management as a way to address technical risk.

In order to formulate a satisfactory design, the decisions that lead up to it must anticipate the future effects. One way to improve the design process is to enable engineers to make better decisions. This idea prompts the development of different modeling methods, which aim to facilitate decision making by using abstractions to represent reality. For different levels of available information, the appropriate modeling method will change, and the challenge lies in selecting the most effective

approach. The setting of this research is at the initial phase of conceptual design, where the system developer has just started gathering information. At this time, all the assumptions are still crude and the system developer has to further investigate the decisions that need to be made. The design of complex systems is naturally challenging, so the goal is not to simplify or automate the decision making, rather to provide a guide that leverages the adaptive judgements of the engineers. As described in a white paper titled “A Complexity Primer for Systems Engineers”,

As the complexity of systems and their contexts has grown, systems engineering methods and tools have increasingly fallen short of what is needed in the face of this reality. A common approach has been to seek clever ways to simplify, or reduce, the subjective complexity so that the problem and the system are understandable. Scientific advances have, in fact, often come from elegant simplifications that model the important variables or forces that dominate behavior. However, this is not always possible – complexity often cannot be simplified away without losing the essence of the problem or possible solutions. Further, this simplification leads to an inability on the part of the solution to be able to engage with the complexity that remains despite our preference to assume it away [4].

Applying these concepts to the assessment of advanced technologies, this study introduces the philosophy of uncertainty management as a way to manage complexity. Advanced technologies bring risks to the program because of uncertainties in their properties and interactions, which can lead to unexpected behavior at the system level that is unknown until the later stages. With the resources and time available during conceptual design, this decision making scenario exemplifies the implication of complexity at the early stage of design. System developers need to determine technical strategies for the program when they have the least available knowledge about the system and the technologies. A decision making framework was developed to support this process as an effort to avert poor system performance or possible program cancellation. This framework incorporates conventional elements of advanced technology assessment and systems development, but promotes an agile attitude to deal with complexity. The goal is to lay a foundation to the thought-process, such that communication can have improved effectiveness by having a systematic look at uncertainty.

It is rarely possible to fully assess in advance what complexity within a project must be addressed and what can be assumed away. Because complex systems often times require novel solutions, systems engineers have to adapt their approach to unfolding conditions. While at its core systems engineering is concerned with the interfaces between and among separable system elements, it should be realized that the more important understanding concerns the dynamic behavior of the interactions between these elements. Properly understood, systems engineering is concerned with context over structure, with interactions over elements, with the whole over the sum of the parts. Moreover, this research recognizes the sociocultural aspect to the design of complex systems. Complex systems demand the collaboration of people from varying disciplines and background. The teams may come from different companies or organizational entities. To work effectively, collaboration includes information sharing, active listening, establishment of trust to enable candid dialogue, and making decisions transparent. A collaborative mindset can lead to deeper engagement practices that enable co-creation systems design.

Chapter 2

UNCERTAINTY MANAGEMENT FOR ADVANCED TECHNOLOGIES

In large aerospace programs, the assessment of advanced technologies typically takes place during conceptual design. This is driven by the need to mature the technology in parallel to the system development. As a result, the system developer needs to deal with the uncertainties that emerge from the low concept maturity. This study presents a decision making framework that implements risk management approach to address the uncertainties in the net potential benefit of the technology.

2.1 Uncertainty as Program Risk

As a brief overview of the terminology that is used when describing uncertainty, there are two common designations: reducible and irreducible uncertainty and/or epistemic and aleatory uncertainty [8]. Reducible uncertainty is associated with things that are knowable but are currently unknown, while irreducible uncertainty is the uncertainty associated with unknowable things. Similarly, epistemic uncertainty, which roots from the Greek word *episteme* for knowledge, refers to the uncertainty caused by lack of knowledge. This incomplete knowledge is a state of the decision maker, rather than a description of the physical system, which can also be termed imprecision. Aleatory uncertainty, on the other hand, is the uncertainty that arises from a random process or inherent variability of a system, stemming from the Greek word *aleator* for dice thrower. The motivation for such distinction is to recognize the notion that there are certain things are inherently unknowable until they are realized, while there are things that are in theory knowable but may be unknown to the decision maker. This research deals primarily with the reducible or epistemic uncertainty, where advanced technologies have characteristics that are currently unknown that may or may not be

favorable at the system level. Figure 2.1 is an illustration adapted from Nikolaidis (2005) that depict the characteristics of uncertainty [9].

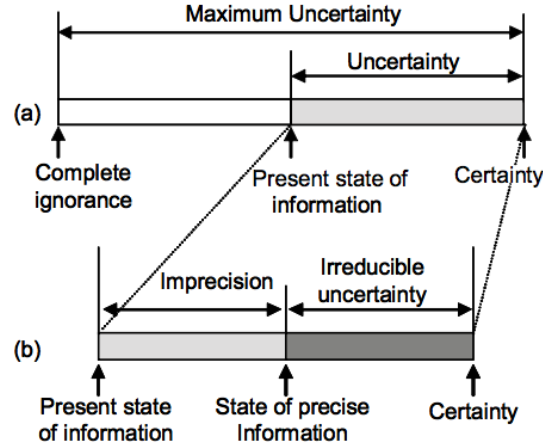


Figure 2.1: State of information, adapted from Nikolaidis [9].

The uncertainty in the behavior of the technology, such as unknown limitation and interaction, leads to the uncertainty in its integrated capability. This implies imprecision in the expected contribution of the technology to the system performance and how design requirements would be met. Uncertainty in the ability to satisfy the requirements signifies program risk. This implies that when dealing with advanced technologies, the program risk not only comes from the technology maturity, but also the from the uncertainty in how much the advanced technology will benefit the performance of the system.

Figure 2.2 is an illustration of the two dimensions of technology assessment. For illustrative purposes, the correlation is shown to be linear, but in reality this relationship can take the form of nonlinear correlation. First, the system developer needs to take into account the maturity of the technology, which can be expressed as Technology Readiness Level (TRL). This can be used to assess whether it can conform to the program timeline. The second dimension that is the emphasis of this study is the uncertainty in the potential net benefit of the technology. Naturally, as the technology matures, the uncertainty in its benefits will be reduced to zero. At the conceptual design phase, however, the system developer needs to deal with the left segment of the diagram. For large aerospace programs, many of the technologies con-

sidered have low TRLs due to the length of the programs, in which the uncertainty is especially high. Consequently, the system developer needs to simultaneously look at the maturity and uncertainty of the technology during the assessment.

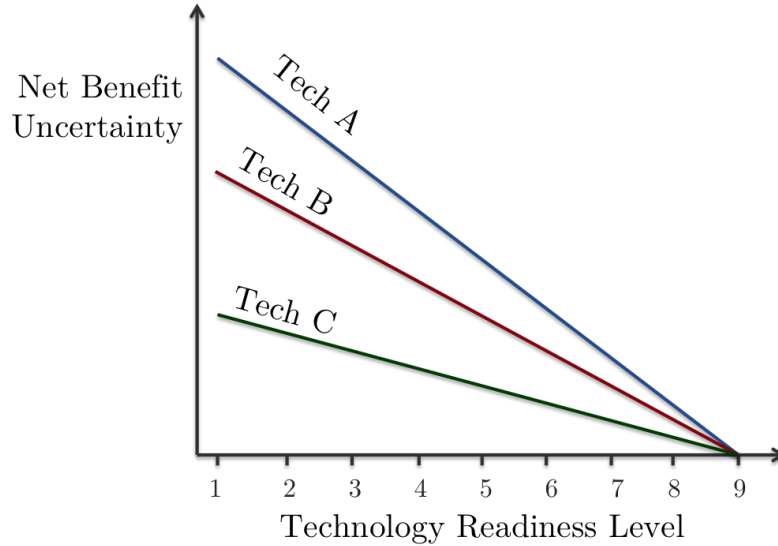


Figure 2.2: Program risk is more than maturity.

When considering uncertainty, it needs be assessed along with the expected impact of the technology. For instance, Technology A can have high uncertainty but is predicted to have high impact on the system performance. Alternatively, Technology B can have medium uncertainty but is predicted to produce low impact on the system performance. This analysis can allow the system developer to explicitly look at the trade-offs between the risk being introduced to the program and the opportunity for delivering a better system. The system developer needs to ultimately determine the expected impact to justify the degree of tolerable uncertainty.

To illustrate the idea of uncertainty, it may be useful to represent the contribution of an advanced technology as the change between a baseline system design, $S_{baseline}$, and the preferred system, $S_{preferred}$, where the variable S expresses any system performance metric. This study is interested in the uncertainty that is present in the ΔS term, denoting the potential system-level benefit. At the conceptual design phase, the most important information to deduce are the types of uncertainties and the impact they can have on the system performance. For example, if there are significant uncertainties in the term ΔS , then it opens the possibility of it having a negligible impact on the system overall.

$$S_{preferred} = S_{baseline} + \Delta S$$

It is acknowledged that the actual net benefit of a given technology will not be known until it is fully integrated into the system. However, the idea is to be able to understand the nature of these uncertainties in order to arrive at the most logical decision. Figure 2.3 shows an illustration of the effects of uncertainty. The typical approach to advanced technology assessment is to focus on the potential benefits. There is a tendency to want to compare systems using specific estimations that the technology developer supplies. This quickly falls short because in reality, the benefit of a given technology can only be assessed in the context of the application. The same technology can play a very different role when integrated into different systems. Depending on where the technology lies in the functional network of the system, the resulting interactions can lead to different emergent behavior. Consequently, there is uncertainty in the net benefit of the technology that needs to be explicitly discussed for the particular application. During conceptual design, quantitative modeling that can effectively capture uncertainties is not possible. Stronger emphasis should be placed in trying to gather information that can help paint the picture. The objective is to consider the uncertainties and their potential degree of impact during the decision making process.

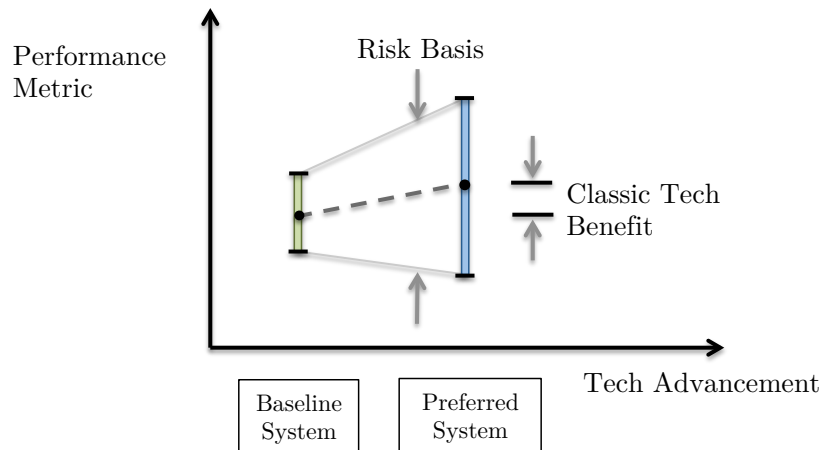


Figure 2.3: Potential implication of uncertainty as performance risk.

To understand what implications the uncertainties have on the program, there are the three levels of uncertainty that need to be recognized when dealing with advanced technologies. The system developer ultimately wants to know the system-level impact of the technology, but needs to confer the technology developer beginning from the from the lowest level.

- 1. Technology:** Uncertainty in how the technology will perform
- 2. Integration:** Uncertainty in how the technology will interact with other parts of the system
- 3. Emergent behavior:** Uncertainty in how the technology will impact the function of the system

Once the system developer gathers information about sources of uncertainty, they can interpret the risk implications at the program level. Program risk can be thought of as originating from three categories: cost, schedule, and performance [6]. Cost deals with the risk of going over the proposed funding, schedule refers to the risk of program delays, while performance is the risk of not complying to the specified requirements. Figure 2.4 illustrates how each element interacts with one another and how it contributes to program risk.

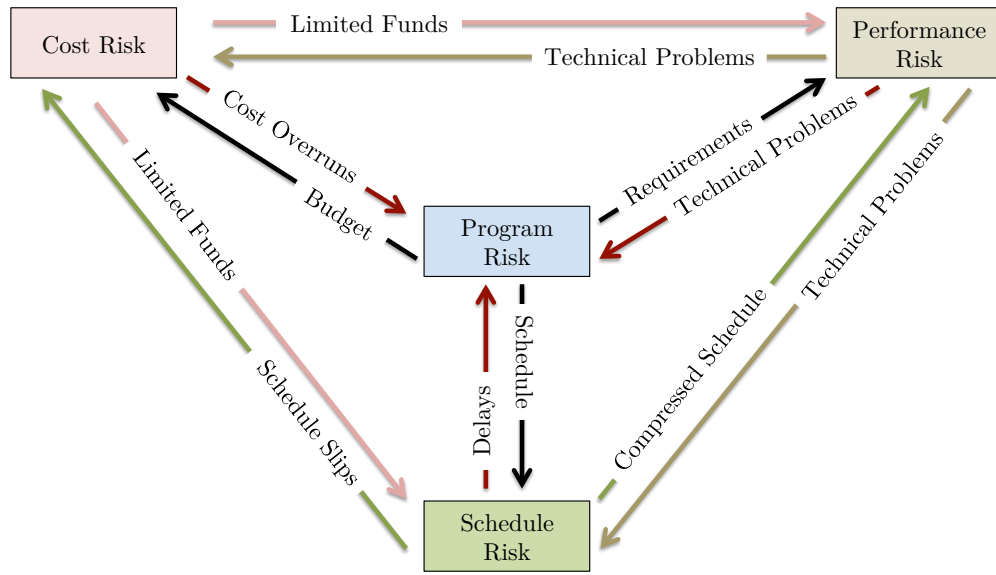


Figure 2.4: Elements of program risk.

Advanced technologies introduce risk in all three categories, where failure to stay on track can have significant impact on the program overall. The questions that need to be addressed include:

How much investment cost is required to develop the technology?

How long will it take until the technology can be fully integrated to the system?

How much net integrated benefit will the technology actually offer?

The uncertainty in the response to these questions directly relates to the knowledge currently available about the technology. There is direct correlation between information and uncertainty, therefore the objective of this research is to maximize the exchange of useful information. Within the short time duration, the system developer needs to gain an understanding of the interplay between what is known and unknown about the technology, and the implications it will have for the program. The study focuses more on performance risk and the ability to satisfy functional requirements, nonetheless, cost and schedule risk are all interrelated. The philosophy is to treat uncertainty as risk and approach it the same way as with risk management.

2.2 Technology Assessment Techniques

In the realm of technology assessment, the most basic approach can be exemplified by the Technology Readiness Level (TRL) [10]. The TRLs are now in common use throughout NASA, industry, and military organizations and are characterized by the ease of classifying a given technology using a set of generally understood parameters. Although it can be a useful method to measure progress toward inclusion, there are a number of limitations that have been recognized with TRLs. One of them is the notion that TRLs do not reflect integration process, which presents a real deficiency in estimating the steps required to achieve flight readiness. Furthermore, it provides little guidance into the uncertainty that may be expected in moving through the TRL levels, as the relative size of TRL steps are not linear and can vary for different technologies. A number of improvements have been developed to address these drawbacks. For example, the System Readiness Level (SRL), which offers a similar structure, adds the integration process to assess the maturity of technology insertion [11]. This technique offers a quick method for categorizing advanced technologies, but a more elaborate process is necessary to assist in the decision-making process that can encapsulate the uncertainties associated with the technology.

On another end of the spectrum, approaches such as technology forecasting use multidisciplinary design optimization (MDO) methods to take a more quantitative approach to assessing technology impacts [7][12][13]. Similarly, the motive is to have a capability to identify high payoff technology areas with minimal investment and resources expenditure to direct research and development resources. The general framework follows a fairly conventional approach, where a baseline system is identified and incremental changes to variables from the design space are applied (i.e. sensitivity analysis). Resulting impacts on previously defined system metrics are quantified using MDO techniques and can later be used as supporting evidence to justify the business case.

It is recognized that different methods have different assumptions and different costs associated with it. Some of these quantitative methods suffer from the assumption of the existence of clearly defined probabilities and well established relationship between design variables. More often quantitative analysis requires a concrete de-

scription of the baseline system, which houses more room for uncertainties that can lead to misleading information. Incautious use of such tools can also cause a false sense of precision. The scaled metric, on the other hand, downsizes the assessment via a binary approach, which fails to capture the full complexity of the problem.

Nonetheless, all of these approaches share the same objective, which is to facilitate the integration of advanced technologies into befitting applications at both the functional and program level. This research aims to complement the existing methods by providing structure to the process that extracts as much useful information about the technology as possible. The quantitative methods can especially benefit from having more informed assumptions. Refined inputs lead to more valuable results. The technology developer can also benefit from having a better maturation strategy. The framework developed in this study can therefore contribute as an addition to the technology assessment methods. To provide a practical tool for conceptual design, the framework implements a qualitative approach as a cheap and easy-to-use method.

2.3 Approach

The main objective of this research is to address the challenges in advanced technology assessment from a risk management standpoint. As with risk, uncertainty is an integral part of the design process, therefore the idea is not to avoid it, but to manage it. To adopt an adaptive mindset, system designers need to be able to appreciate and tackle unpredicted behavior as it appears. In this study, the approach is to guide the qualitative discussion of uncertainties at the earliest phase of the design. This can help prepare the system developer to navigate through the complex reality of the process. The underlying goal is to expound the principle of uncertainty management and to embrace complexity by trying to understand the significance of the unknown characteristics.

It was established that uncertainty in the net potential benefit signifies risk to the program; as a result, the system developer needs to assess the level of risk associated with any technology being considered. Figure 2.5 shows the basic process for risk management, where the ultimate objective is to make informed decisions by consciously assessing potential losses and the resulting impact. Employing the same

concept, uncertainty in advanced technologies needs to be identified and analyzed to substantiate the design decisions and program planning.



Figure 2.5: Risk management process [14].

The first step in managing uncertainty is to identify the sources of uncertainty. This process involves the current state of information about the technology and how it relates to the system application. This alone is not enough since it does not determine the extent of the risks. Certain questions will remain unanswered, such as the likelihood of unfavorable outcome and the potential impact on the program. In effect, the identification of the most serious or important uncertainties enable them to be prioritized. Once the uncertainties are characterized, the extent of the risks can be evaluated more readily. This can then feed into the development of action plans to prevent the occurrence or minimize the impact of the consequences.

To guide the risk assessment, a decision making framework was developed to structure the interaction between the system developer and technology developer. The framework focuses on facilitating the collection of information so that both groups can gain insight on the characteristics of the technology integration. High-level tasks were identified that encompass the objective and suggestions for implementation. This is meant to reduce the noise in the information exchanged. The key considerations include technology applicability, integration effects, maturation milestones, system interaction, uncertainty characterization, and lastly risk implications. Once information relating to these factors is gathered, a strategy for decision making and program planning can be made. The following are some of the motivating questions in this study:

- How should technology maturity and applicability be assessed by the system developers?

- How can system developers identify all the subsystems that will be impacted by the technology early on?
- How should system developers plan for technology development to ensure compliance with requirements?
- How should system developers manage information to support decision making under uncertainty? What information is critical/necessary?

This framework is designed to provide a more comprehensive landscape of the technology. By ushering in the conversation of uncertainty early on the design process, the system developer can more effectively evaluate the risk and opportunity trade-offs. The knowledge-based approach is essentially centered around building a practical connection between the vision of the technologist and the practical benefit the system developer wants. If both groups can establish a solid strategy for the technology integration, it will help create a synergistic effect that enhances the overall quality of the system. The framework aims to provide a foundation to the overall thought-process and is intentionally designed to be open-ended. To deal with complexity, the decision making ultimately relies on the engineering judgment of the decision-makers. The flow is simply to guide the interaction between the two groups involved in the process, such that information is relayed more systematically. At the end of the framework, the system developer will gain a more comprehensive look at the implications of the technology inclusion.

Chapter 3

EXAMPLE CASE OVERVIEW

To illustrate how the decision making can be applied in practice, a technology assessment using actual advanced technology research was done as part of this study. This was employed to an example of a system development program. The example is used as a platform to explain the thought process behind the decision making framework, which is discussed throughout the next chapter.

3.1 System

In response to growing interest in unmanned systems, there is increased effort to expand the application domain. A project was initiated by a start-up company whose objective is to transfer the capability of high-altitude platforms to support the commercial sector. The system being considered is a Tier II+ class unmanned aircraft system (UAS). This is used as an example of a system development program.

The goal of the program is to provide persistent coverage that supports observation and/or communications relay missions. The end-user was identified to range from telecommunications companies to natural disaster response agencies. In the commercial sector, continuous coverage has been largely supported by the use of satellites. While this has been an adequate solution, the cost associated with developing satellites are relatively high, and they have rigid capability once it is in orbit. Additionally, it is much more difficult to rectify problems during operation due to limited access to the spacecraft and its surroundings; if the spacecraft experiences problems, it cannot be retrieved and can not have any maintenance work to prolong its lifetime [15].

To address some of these challenges, the alternative solution being considered is to use a high-altitude, long-endurance vehicle. While providing continuous service like satellites, HALE UAS offers an added value from its versatility. Similar to aircraft operations, HALE UAS can be deployed quickly and be recovered for maintenance. The vehicle can support broader types of mission from being able to swap payloads depending on the user needs. In comparison to satellites, this alternative can also offer better performance capabilities from the lower altitude, such as data resolution and power requirement.

Based on these motivating ideas, the system developer aims to develop a cost-effective solution that fills the gap between aircraft and satellite. Primarily intended to perform the missions of existing satellites, the system strives to bring in the advantages of aircraft operations, particularly in terms of flexibility and quick deployment. The system being developed takes inspiration from existing vehicles, such as Lockheed U-2 Dragon Lady and Northrop RQ-4 Global Hawk, which are mostly used for research or military support. The system developer plans to offer better performance, such as improved endurance, while targeting the commercial market.

Figure 3.1 shows the basic timeline for the program, which encompass a 10-year period. The program is currently at the conceptual design phase. To provide the required continuous service, it was determined that a minimum of two vehicles is needed to provide persistent coverage that includes transit, loiter, and standard turn-time for post-flight inspections. The design mission comprises an endurance of 31 hours at a cruise altitude of 55,000 - 65,000 ft. From consulting potential customers, the system developer specified a minimum payload capacity of 3,000 lbs.

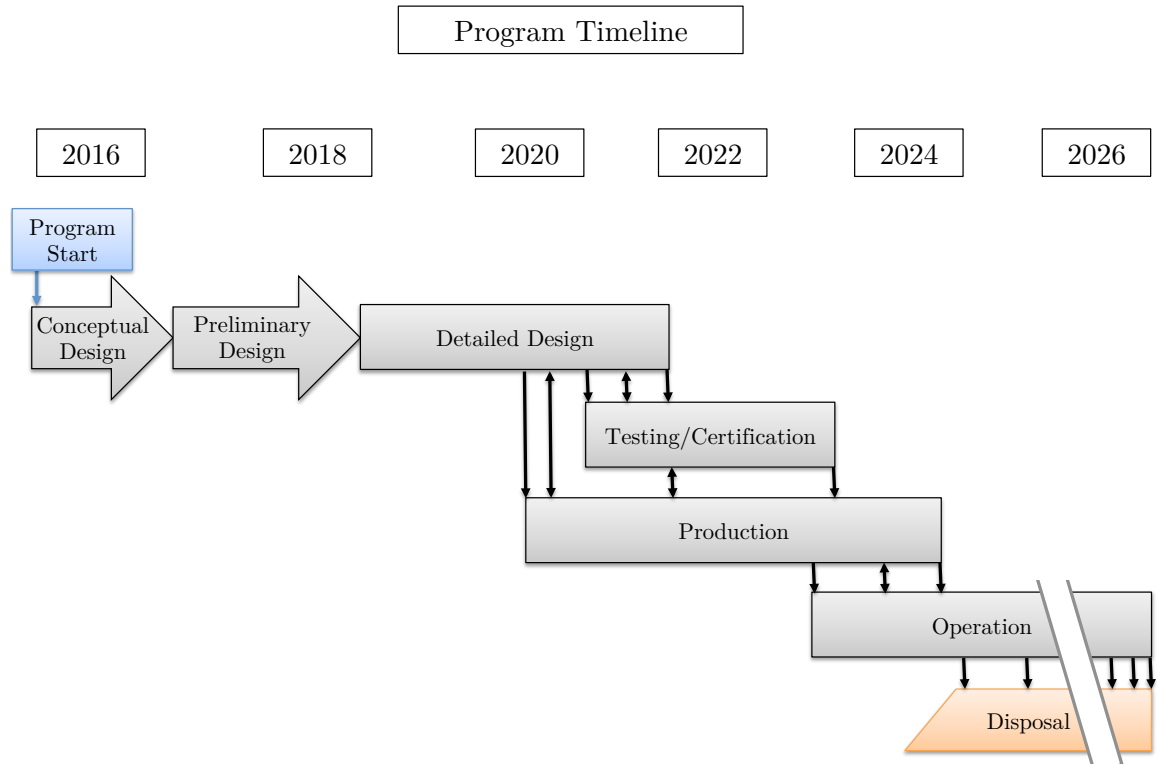


Figure 3.1: Program timeline.

At this stage, an aircraft sizing iteration was done to determine the basic characteristics of the vehicle. The assumptions used for this calculation are listed in Table 3.1 and the resulting system attributes are summarized in Table 3.2. This was done to illustrate the state of information that the system developers typically have at the time where the decision making framework is most relevant. The decision making framework is tailored for the beginning phase of the design process. During conceptual design, the system developer is not expected to have refined description of the system before consulting technology developers. It is assumed that the system developer performs the technology assessment concurrently to the initial design iterations.

Table 3.1: Initial assumptions

System Metric	Value
Payload	3,000 lb.
Endurance	31 hrs.
Weight Fraction	0.36
Lift-to-Drag Ratio	25
Cruise Condition	M0.6 at FL550
Specific Fuel Consumption	0.6
Lift Coefficient	0.7
Aspect Ratio	22
Taper Ratio	0.25
Reynolds Number	2.6506×10^6 at M0.6/FL600
Airfoil	HSNLF(1)-0213

Table 3.2: Initial aircraft sizing

System Metric	Value
Max. Takeoff Weight	26,040 lb.
Empty Weight	9,375 lb.
Engine Thrust Requirement	$2 \times 1,040$ lbf.
Fuel Fraction	0.525
Wing Area	691.7 ft ²
Span	123.4 ft.
Root Chord	9.0 ft.
Tip Chord	2.3 ft.
Mean Aerodynamic Chord	6.1 ft.
Leading Edge Sweep	6.22°

Only one element of this system design is closely examined in the example case. The system developer is interested in selecting a technology that promises the highest payoff while imposing acceptable risk to the program. As a HALE UAS, one of the design objectives is to maximize the endurance. This is used as basis for the technology assessment discussed in the subsequent chapter.

3.2 Advanced Technology

In conjunction to the system development program, an advanced wing design was selected as the technology example. The advanced technology is based on a research project conducted at NASA Neil A. Armstrong Flight Research Center. The background and research findings were documented in the paper titled, “On Wings of the Minimum Induced Drag: Spanload Implications for Aircraft and Birds” (2016) [16]. The research study was primarily based on the implementation of an alternative solution to the Prandtl elliptical spanload distribution, referred to as the bell spanload. Three sub-scale flying wing aircraft have been made during the project that used wing twist to achieve the selected bell shaped spanload. The result of this research proposes a wing design for improved aerodynamic efficiency (i.e. drag-reduction technology).

Until now, Prandtl’s lifting-line theory (1920) remains a standard tool for understanding and analyzing aircraft wings. This theory points to a solution for the most efficient wing, which takes form of an elliptical spanload. However, little-known paper by Prandtl (1933) presents a superior spanload that has a lower induced drag solution than the elliptical spanload with a given structure. The research project aims to provide experimental validation to the aerodynamic solution described in the paper, which is claimed to correspond to bird’s flight dynamics. Figure 3.2 shows a schematic that compares the traditional elliptical spanload to the bell spanload [16].

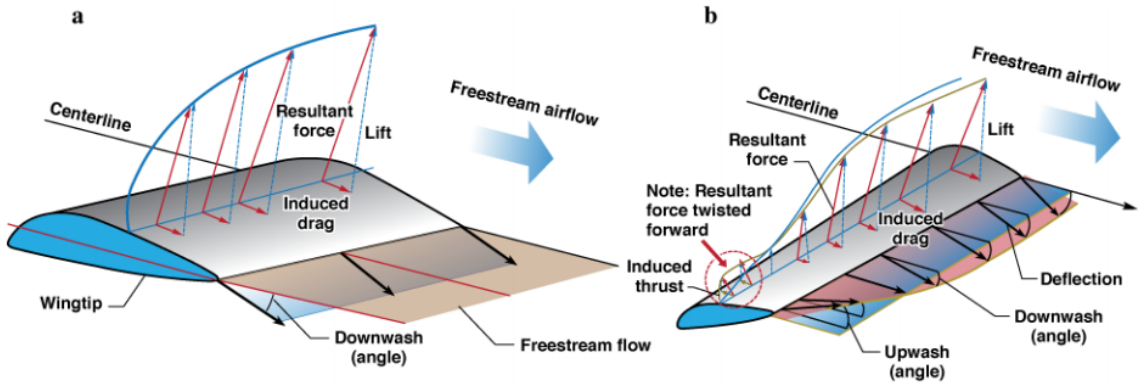


Figure 3.2: Prandtl’s elliptical and bell spanload explained [16].

The research includes an experimental comparison with avian data. The paper suggests that having no other model, avian researchers have used the elliptical spanload virtually since its introduction. Yet over the last half-century, research in bird flight has generated data that was used for comparative analysis. The research argues that the bell spanload is the correct model for bird flight data:

“Based on research we present a unifying theory for superior efficiency and coordinated control in a single solution. Specifically, Prandtl’s second spanload offers the only solution to three aspects of bird flight: how birds are able to turn and maneuver without a vertical tail; why birds fly in formation with their wingtips overlapped; and why narrow wingtips do not result in wingtip stall” [16].

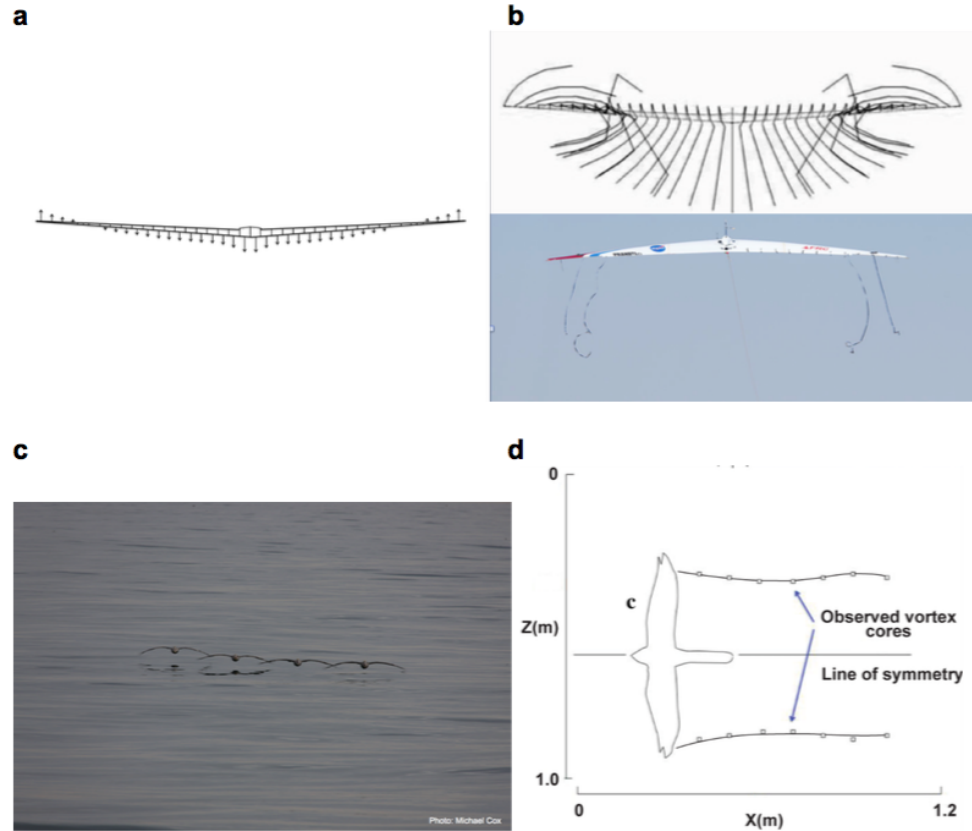


Figure 3.3: Bell spanload wingtip characteristics. (a) Spanload downwash profile (b) Vortex rollup behind the wing, analytical and in flight (c) Formation flight of Brown Pelicans demonstrating wingtip overlap (d) Spedding’s Kestrel data showing inboard vortex core location[16].

Three experimental aircraft were designed in accordance with Prandtl’s 1933 paper to validate the various potentials of the new spanload. The vehicles implemented the bell spanload solution to verify the consistency with avian data, to demonstrate the concept of proverse yaw, and to offer a new method of aircraft control and efficiency. The model planform was a 25-percent Horten H Xc aircraft (12.3 ft span) with a design lift coefficient of 0.6. The elevons had equal and opposite throws while functioning as ailerons. The demonstrator had no differential bias, meaning it used a direct, stick to surface control system [16].

Table 3.3 summarizes the characteristic differences between the elliptical and bell spanload. The bell spanload coordinates the roll-yaw motion and maximizes aerodynamic efficiency with a given structure. The paper states that these findings allow for improved aircraft designs, particularly all flying-wing aircraft and blended-wing body aircraft [16]. This technology is considered for a conventional tailed aircraft to improve the aerodynamics performance. Since the system is a HALE UAS, there is less restriction on the span and the minimum structure approach can potentially provide better solution. The primary implication on the design of the wing would be the implementation of a wing twist distribution (provided in the research paper) that produces the bell spanload.

Table 3.3: Characteristics comparison of elliptical and bell spanload [16].

Flow Field Parameter	Elliptical Spanload	Bell Spanload
Lift	Remains large from root to wingtip	Tapers from large at root to zero at wingtip
Slope of lift at wingtip	Undefined at wingtip	Tapers to zero at the wingtip
Downwash	Constant across the span, piecewise discontinuous at the wingtip	Large at root, rises to zero at 0.704 span, and upwash at the wingtip
Wing vortex	At the wingtip, strong and uncontrolled rollup, vortex is tight and strong	Located at 0.704 span, controlled and weak, vortex is smooth and weak
Upwash region	Exists only beyond the wingtip (outside the vortex), small and turbulent	Begins at 0.704 span, large and smooth controlled

Chapter 4

DECISION MAKING FRAMEWORK

In this framework, the assessment of advanced technology is addressed in three folds, namely applicability, maturity, and risk implication. Shown in Fig. 4.1, the framework specifies high-level tasks for the system developer, the technology developer, and both. The process will naturally involve the engagement of both groups in all the steps, but the tasks are compartmentalized to highlight where the information and decisions should be coming from. Analogous to a work breakdown structure, this framework aims to decompose the decision making process into manageable sections. The system developer and technology developer have differing perspectives on the effects of a technology, so the approach was to establish objective-oriented steps that outlines the discussions to enhance the quality of the information being exchanged.

This decision making framework is not intended to reduce the complexity of the problem, but rather structure the interaction to help gain informative insights with the current state of knowledge. The net benefit of a given technology can only be assessed for the application it is being considered for, meaning the value of its function is system-dependent. To arrive at a good decision, both groups need to collaboratively exchange ideas to establish a foundation for the applicability to the specific program. This socio-technical process is often overlooked, but is where many problems are sown due to the risk involved with advanced technologies.

With the limited resources and time during conceptual design, system developers generally want to minimize making conjectures that can hurt the program later. It is therefore critical for the system developer to have a systematic approach, especially for larger programs that have additional factors that contribute to the complexity. This framework is tailored for conceptual design by emphasizing high-level discussions that

will depict a more comprehensive look at the implications of the technology inclusion. The knowledge gathered through this process can facilitate prioritization to improve resource management. The subsequent sections elaborate on each task, where the example case is used to demonstrate the implementation.

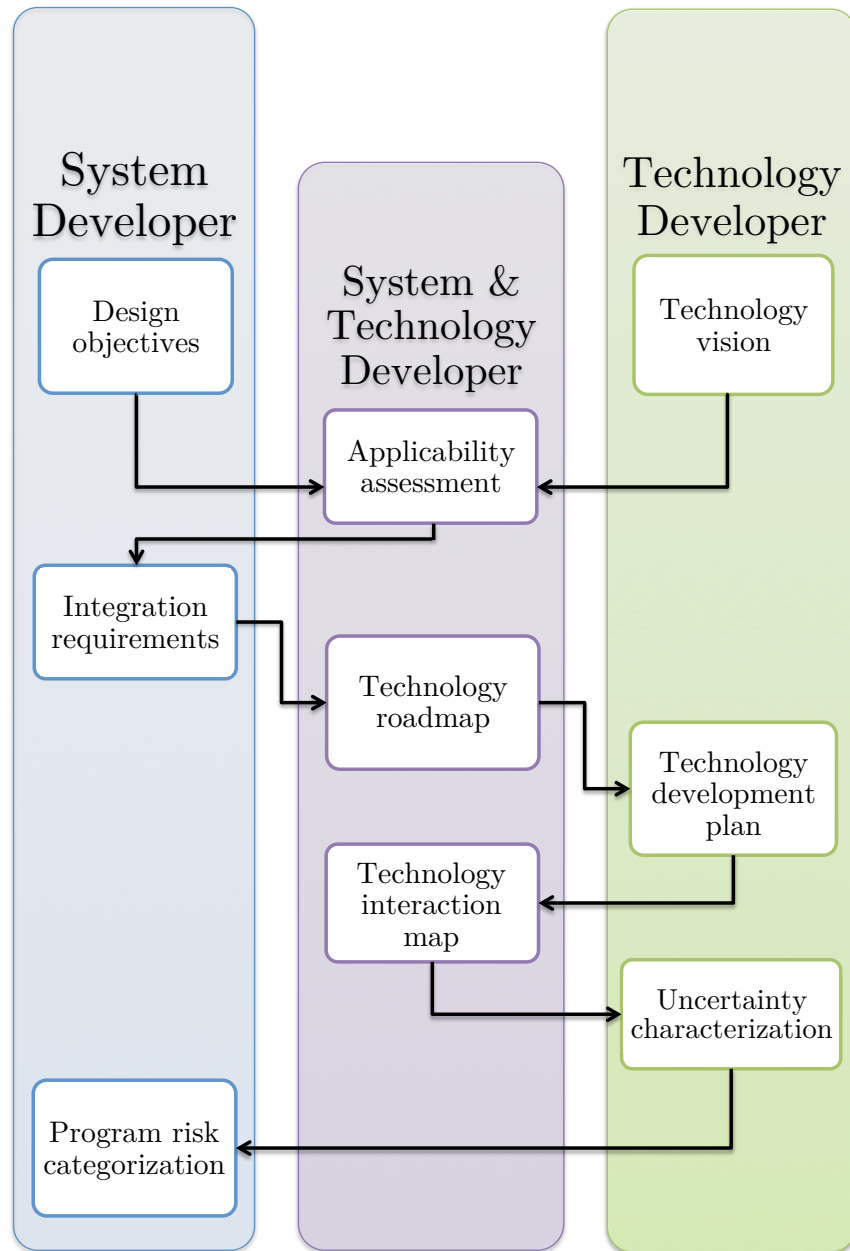


Figure 4.1: Technology assessment framework.

4.1 Design Objectives

In order to select an appropriate technology candidate, the system developer needs to first establish specific design objectives that the technology is intended to address. This represents the problem formulation phase in decision making. As discussed in Chapter 2, the quality of a solution is closely linked to the problem definition. The design objectives need to be well thought-out to express system performance targets that are critical to the success of the program.

To arrive at a sound design objective statement, the system developer needs to interpret the requirements and have a thorough understanding of the desired functionality of the end-user. It is assumed that the system developer has already performed user needs analysis, and have transformed this into system-level and derived requirements. An understanding of the factors that are driving the design is also expected to provide basis for evaluating the opportunity associated with different technologies.

At this step the system developer needs to identify a specific performance metric or design variable to focus on. This facilitates the process of reviewing technology portfolio and selecting a technology that can be clearly linked to the design objective. The design objective is central in the applicability assessment process of the technology and therefore should be considered as a preliminary step to understanding the potential benefit of a given technology.

Moreover, the performance objectives must be meaningful to the system developer so that the implications of meeting or not meeting them to various degrees can be understood. The system developer must ultimately be able to articulate a rationale for preferring one decision alternative over all others, which requires some ascribed value, at least qualitatively, to the degree to which the various alternatives meet the performance objectives.

Once the design objective is identified, this is used to select advanced technology candidate. The system developer can assess advanced technology portfolio and review the stated objectives of each technology. If the company does not have an internal technology portfolio, then the process involves researching different advanced

technology projects that are aligned with the design objective. The system developer then approaches the technologist to inquire more information about the technology.

4.1.1 Example Case

To address the functional need for a system that can provide continuous coverage, the design strategy was to develop a vehicle that has the capability of a Tier II+ UAV but is configured for commercial missions. The Tier II+ classification refers to a UAV that operates at high altitude (60,000 to 65,000 feet AGL) and has long endurance (at least 3,000 nm radius and 24hour time-on-station). The system will comprise multiple vehicles that will perform the mission in rotation.

The design objective for the advanced technology assessment is to address the 31-hour endurance requirement for the vehicle. Since the system needs to act as a telecommunications relay or observation platform, maximizing the time on station will add value to the system's competitiveness. From the aircraft design perspective, there are a few different ways to improve endurance. The Breguet range equation describes the relationship between some properties and the range of the vehicle.

$$Range = \frac{V}{g} \frac{1}{SFC} \frac{L}{D} \ln\left(\frac{W_{initial}}{W_{final}}\right)$$

Starting from this elementary standpoint, range can be increased by the following ways:

- Decrease specific fuel consumption (SFC)
- Increase lift-to-drag ratio (L/D)
- Decrease weight fraction ($W_{initial}/W_{final}$)

The system developer can subsequently examine the significance of each option. The first option implies the improvement of the engine efficiency. While performance requirements for the engine will be specified, this option was considered more limiting since the engine will be largely driven by the need to operate at high altitudes. The second option suggests improving the aerodynamic efficiency of the wing and overall

vehicle shape. This requires the system developer to allocate resources to perform extended analyses to improve the aerodynamic design of the vehicle. The third option proposes that the fuel fraction be improved by driving the aircraft empty weight down. This is the ideal strategy to make the overall system better, but is in reality the hardest challenge due to the uncertainty in the components that will make up the aircraft weight and their mass properties. The overall design process will inherently aim to minimize the weight of the aircraft, so this option was considered to be less tangible for a single advanced technology to address. For the first option, adding a requirement on the fuel consumption of the engine can lead to over-constraint design because the engine already needs to operate at high altitude. Since HALE UAVs spend significant portion of their mission in the cruise segment, it was concluded that improving the aerodynamic performance would be a sound strategy with the resources required.

4.1.2 Key Idea and Applicability

This first step is mainly to prompt the system developer to come into the process with a clear objective. As discussed in Chapter 2, the problem formulation is equally as important as the problem solution. The system developer needs to have a good understanding of the capabilities that are critical to the end-user to decipher how the requirements shape the design space. The system developer needs to subsequently assess which system performance metrics can leverage the value of the system. A starting point would be the figures of merit, but the system developer needs to try to gain a deeper understanding of the system requirements to be able to define the level of acceptable risk accordingly. Higher risk can be justified more logically if there is potentially a high payoff, but not when the expected impact is not significant. The design objectives should be able to reflect this value in terms of system performance.

For large aerospace programs, advanced technologies play an important role in helping to answer the feasibility question. Having a clear design objective is therefore pivotal to identify the potential contribution of the technology in the system. The business case for advanced technologies need to be supported with a cogent explanation of how they will add value to the system being designed.

4.2 Technology Vision

As an entry point for the technology developer, this step is intended to provide a closer look at the potential benefits and envisioned application from the perspective of the technologist. The goal is to give an overview of the motivating ideas behind the technology and the work that has been done so far. This should give the system developer some background information to understand the relevance of the technology in the context of the system being designed.

This step should include an explanation of the physical phenomenon being exploited, as well as some historical perspective on the progression of research in this field. The technology developer needs to also describe the significance of learning more about the topic and vision for the technology, such as implications on other research fields. While this is not a sales presentation, this is a chance for the technology developer to discuss the potentials of the technology, share the research findings thus far, and the imagined future for its applications.

The system developer should frame the conversation such that the technologist is encouraged to talk about the state of knowledge concerning the technology. While it is important to know the information that has already been discovered through the present work, it is even more essential to talk about what is still unknown about the physical phenomenon and/or technology. The discussion of uncertainties should begin as early as this initial research phase; the earlier the uncertainties are made transparent, the better understanding the system developer can have about the risk that is involved. This can be initiated by talking about the challenges that the technology developer is currently facing, such as the gaps in the current knowledge or technology resources.

4.2.1 Example Case

Preliminary Research Aerodynamic Design to Lower Drag, or Prandtl-D, is a project being done at NASA Armstrong Flight Research Center that features a new method for determining the shape of the wing with a twist that could reduce induced drag [17]. The chief scientist, Albion Bowers, leading the research project was primarily

motivated by the second paper written by Prandtl in 1933. The theory discussed in the paper showed that there is another solution for the minimum induced drag other than the elliptical spanload. The research objective was to initially provide experimental validation that the bell-shaped spanload distribution produces proverse yaw at the wingtip and is the correct model for the flight of birds. The technology developer believes that this can have a far-reaching impact, “In aircraft alone, should the complete drag reduction be taken advantage of, 11% reduction in total aircraft drag due to spanload, elimination of the tail results in another 20-30% efficiency gain, and then 15.4% improvement in propulsive efficiency, the total efficiency increase is on the order of 60%, a substantial reduction in carbon footprint” [18]. The research project aims to advance the experimental analysis to show this long-term potential of the technology.

Two sub-scale demonstrator vehicles were previously built (12.5 ft wingspan and weighs 15 lbs) to analyze the yaw moment characteristics and vortex roll-up location. This was considered a “control or flight mechanics tested”, instead of a setup that tests the aerodynamic performance [19]. The wings were designed using nonlinear twist and airfoils that vary linearly from the centerline to the tip to produce the bell spanload, and were constructed of a machined foam core wrapped in a skin of carbon fiber. The experimental data suggested that the bell spanload has 11% lower drag than the equivalent elliptical spanload wing, with 22% more wingspan but using the same amount of structure. The research proved conclusively that proverse yaw exists and that bell spanload is the only viable solution that explains why birds can coordinate the roll-yaw motion without vertical tail and why the birds fly in formations with their wingtips overlapped. The difference between the bell spanload and elliptical spanload is summarized in Table 3.3 [16]. The research group has developed a third vehicle, Prandtl-D No. 3, that is approximately twice as big as the previous demonstrator with a wingspan of 25 ft and weighs 28 lb. The Prandtl-D No. 3 first flew late October 2015. It uses the same flight control system as the second Prandtl-D subscale model and is constructed of carbon fiber, fiberglass and foam [17]. Currently, the technology developer is in the process of preparing for the next set of flight test, which will incorporate a pressure-sensing instrumentation to study the aerodynamic performance.

The overall vision is that the bell spanload can be used for improved aircraft design, particularly all flying-wing and blended-wing body aircraft that experience design challenges in the area of yaw control and stability. Conventional aircraft can also benefit from improved aerodynamic efficiency. When implementing the minimum structure approach, the penalty is mainly in the wingspan, which can be a constraint for some aircraft. The goal is to advance the research to study the aerodynamic performance at operating conditions that more relevant for aircraft design, until it can be integrated into full-sized aircraft [16].

4.2.2 Key Idea and Applicability

The framework recognizes that this process involves two groups of people that have differing perspective on the effects of the technology. The technology developer is very familiar with the motivating ideas of the research, but might be less in tuned to the specific systems that could gain practical benefit from the technology. As a starting point for the technology developer, this step is about discussing the envisioned applications and the current state of knowledge. This will help the system developer gain an insight on the environment that the technology is living in, such that the applicability can be assessed for the system being designed.

4.3 Applicability Assessment

The first element to advanced technology assessment is applicability. This can be considered the opportunity evaluation stage, where the system developer and technology should discuss how the technology inclusion can benefit the system. The main task in this step is to explicitly establish the relevance of the technology to the system by correlating the design objective and technology vision. Using the information from the previous steps, the system developer should assess if the system aligns with the envisioned application of the technology, and if the potential benefits align with the system performance objectives.

The system developer can begin by providing a description of the system being designed and why the technology was initially considered. This should involve some discussion about the expected relationship between design variables and the potential role of the technology in the program. The technology developer should concurrently assess if the system has the relevant operating environment initially envisioned for the technology. The applicability can be justified if there is clear compatibility between the performance objectives and the technology.

The discussion should also encompass the ‘best case scenario’, where both the system developer and technologist identify specific ways that the technology can add value to the system. An emphasis should be made on the assumptions that go into the promised benefit, such as the factors that play a central role in the performance of the technology and how the predictions were calculated. The system developer can make a more objective evaluation by maintaining an awareness of how the assumptions relate to the stated benefits.

Along with understanding the upper limit of the technology benefit, another important feature to discuss is the external factors that influence the technology maturation. This can be initiated by talking about dependence on other technologies and resources critical to the development. The system developer should interpret this as sources of risk and determine if they impact the applicability of the technology to the specific program (e.g. can the program provide the necessary resources or build in schedule slack for the dependencies).

4.3.1 Example Case

From the earlier step, it was determined that the design objective for the HALE UAV is to maximize the endurance by improving the aerodynamic performance of the vehicle. Any drag reduction technology can be considered for this objective. The Prandtl-Bowers wing was selected as it applies a more passive approach, involving the design of the wing, rather than an active control device. It proposes the use of a bell-shaped spanload profile in the wing design for lower induced drag. This could be especially beneficial for the application since the HALE UAV will spend a significant portion of the mission in the cruise segment, in which the majority of the drag comes from induced drag. Improved aerodynamic efficiency will ultimately lead to longer endurance.

The Prandtl-Bowers wing has a few built demonstration vehicles, indicating some level of concept maturity. The experiment suggested that the bell spanload could provide “up to 11% drag reduction, at the cost of 22% more span than the equivalent elliptical-shaped wing but using the same amount of structure” [19]. While the integrated benefit of the technology will lead to a different percentage of drag reduction (or potentially drag increase), this system is applicable because the penalty on span is less of a concern for HALE UAV. Since the vehicle will be operating at very high altitude, the wings need to be relatively large to begin with and does not have a span constraint as with large transport aircraft. One of the early consideration was the uncertainty in scalability effects. Although the bell-shaped spanload is argued to be the correct basis for the flight of birds, birds only fly in low Reynolds number, thus it is inconclusive whether the same performance benefit will be observed at the operating conditions of the aircraft.

4.3.2 Key Idea and Applicability

The concept of designing for synergy is the underlying motive for this step. When the envisioned application coincides with the design objective, there is greater potential for the interaction to be favorable to the system network. The objective is to formally establish the relevance of the technology to the system being designed. If the technology vision does not match the design objective, the technology assessment can be

concluded at this step. If there is a clear connection between the two, the system developer should elaborate on the role that the technology will have in the application. The applicability needs to be assessed in terms of the relation of the technology to the system (i.e. envisioned application and design objective) and also to the program (i.e. current level of maturity, external factors, and required resources). If the technology exhibits compatibility in this consideration, the assessment can proceed to discuss integration effects.

4.4 Integration Requirements

Once the applicability is established at the highest level, the next consideration is to understand the environment within the system that the technology will exist in. The system developer should provide a more detailed description of the system being designed, such as the system architecture that the technology will need to integrate into. This can be interpreted as the system boundaries that the technology will need to consider during maturation. As previously discussed, many challenges concerning advanced technologies arise from the need to develop both the system and technology in parallel. There is risk that comes from uncertainties in how the two will interact and what integration challenges will result from these interactions. At this stage of the design, the most important goal is to arrive at the same level of understanding of the integration effects. This can initially be done by examining the connectivity of the system and technology.

The system developer and technology developer need to first establish the functions that the technology will represent, and determine the parameters that the technology will have influence over. It can be useful to explicitly draw the boundaries of the technology and identify other functionality that the technology will exist with. The system developer needs to relate any system-level requirements that are also within the system boundary that the technology inhabits.

4.4.1 Example Case

Figure 4.2 is a sketch of the system boundary for the wing design. The idea is to initiate the discussion on high-level interactions between the technology and the system to help understand the potential integration effects. The technology can be treated as a black box, and the focus should be on identifying all the other parameters that the technology will have to interface with. As shown, the wing design will need to coexist with other functions of the vehicle, such as the flight control system (i.e. control surfaces). The system developer needs to consider how the technology will fit into other components of the system to anticipate the integration process. The integration requirements are meant to ease this process in the future by outlining the environment of the technology within the specific system.

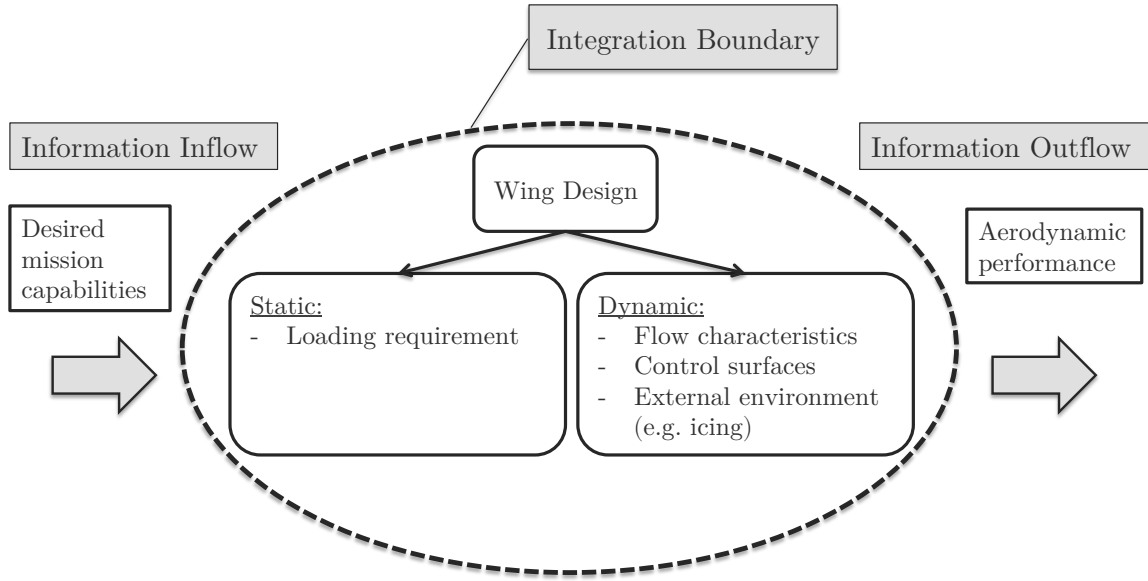


Figure 4.2: Example integration boundary for the advanced wing technology.

For example, the operating conditions of the HALE UAV require the system to have anti-icing capability. While the design of the ice protection system is not critical during conceptual design, the system developer needs to address all the factors that can affect the integration process. Considering that the anti-icing equipment will be manufactured by another supplier, the fabrication of the wing will eventually have to deal with integrating these other devices. The technology developer needs to be made aware of all the other functionality that the technology will need to coexist, such that inputs feeding into the design will be more complete. This is translated into an integration requirement for the technology developer, where the design of the wing must incorporate some means of ice protection system.

Another example of integration requirement is from a design perspective, where the system developer specifies a performance objective for the technology. For the Prandtl-Bowers wing, one requirement can be that the wing needs to have laminar flow during cruise conditions. This integration requirement exemplifies how the system developer places a stipulation on the technology performance to help ensure that the technology will have a constructive contribution to the system. At this step,

there are two types of integration effects that need to be examined, the physical and non-physical. The performance requirement exemplifies an integration requirement for the way the technology will influence the information flow within the system. Alternatively, the aforementioned requirement is an example specifying a physical connection that will be required for integration. The system developer and technology developer need to communicate these relationships early to anticipate the integration challenges and focus the planning efforts accordingly.

4.4.2 Key Idea and Applicability

The primary goal of this step is to begin assessing integration effects. Many integration challenges originate from the need to combine system elements that have different functionality. With limited knowledge of their properties, the system developer needs to merge the system components together in hopes for a synergistic behavior. Coming up with integration requirements is a way to evaluate what is already known about the system architecture and examine the expected interaction. The technology developer benefits from understanding other functionality the technology needs to coexist with, as well as the performance objectives, to come up with a better strategy for the technology development. On the other hand, the system developer can gain a better understanding of the implications of selecting a given technology from the integration perspective.

4.5 Technology Roadmap

One element of any successful program is good planning. Although staying on schedule is always a challenge, large programs depend on being able to keep track of progress and having milestones to work toward. In the context of advanced technologies, it is even more critical to have a thorough plan considering the uncertainties involved with materializing new ideas. Moreover, there is less room for trial period and additional testing when advanced technologies need to be developed in parallel with the system. This step is intended to address the planning of the technology development that is specific to the program.

The primary task is to define specific milestones for the technology maturation and assess how it will fit into the program timeline. Many of the advanced technologies considered in conceptual design for large aerospace programs are in the lower TRLs, therefore it is important for the technology developer to have a time frame to work towards. The technology roadmap should aim to provide specific targets to show the maturation process. This can refer to a series of technology demonstrations necessary to lead up to the final assembly. This step should serve as a platform for discussing feasibility. More importantly, if the technology has dependence on other technologies, it is critical for the technologist to explain the implications of the dependence on the schedule and what type of effects can it have on the technology being considered.

4.5.1 Example Case

Figure 4.3 shows an example timeline for the HALE UAV program. The system developer can determine how much detail is necessary for the timeline, but the goal is for the technologist to get a sense of the time frame that the maturation process needs to conform. The system developer should consider key capabilities that must be demonstrated from the perspective of the required performance to help determine the most appropriate requirements.

Working from the program timeline, the system developer and technology developer need to brainstorm what approach to take to mature the technology. Both sides should provide feedback on what constitutes a sensible target and what demonstration

is appropriate. The system developer needs to determine the features of the technology that would be of interest in the different design phases, such as information that is necessary for a particular analysis. Although the specifics of the design analysis are still unknown at the conceptual design phase, the system developer should evaluate the progression of knowledge that is necessary. The technology developer needs to then respond with ideas for demonstrating the progress.

Figure 4.4 shows an example technology roadmap for the Prandtl-Bowers wing. This technology is currently at TRL 3 and is expected to reach TRL 7 during the detailed design phase, where it would have undergone wind tunnel testing and two major demonstrators (subsonic light aircraft and transonic full-scale aircraft). The first demonstration milestone is tailored to the current project that the research team is doing, in which they are continuing to study the aerodynamics of the wing design. The research group is planning to install pressure-sensing instrumentation in their third vehicle, Prandtl-D No. 3, and begin another series of flight testing this year (2016) [19]. This is an example where the system developer starts by orienting the demonstration requirement with the current work that is being done. When advanced technologies are considered at the start of the program, the technologists may be in the midst of their research work. The system developer should try to smoothly integrate the technology into the program by determining initial demonstration requirements that align with the current research. The demonstration phases can subsequently transition into being tailored to the specific application.

This is an exercise for both groups to exchange ideas on the type of information that would be necessary as the program progresses, which should be the basis for all the demonstration activities. In the example case, for instance, it is desirable for the system developer to refine the lift-to-drag ratio assumption as the program goes through preliminary design. Therefore, there needs to be technology demonstrations that focus on analyzing the aerodynamic performance of the wing design, such as wind tunnel testing. Technology demonstrations are done to reduce uncertainty and needs to be tailored to provide meaningful knowledge to both the system developer and technology developer.

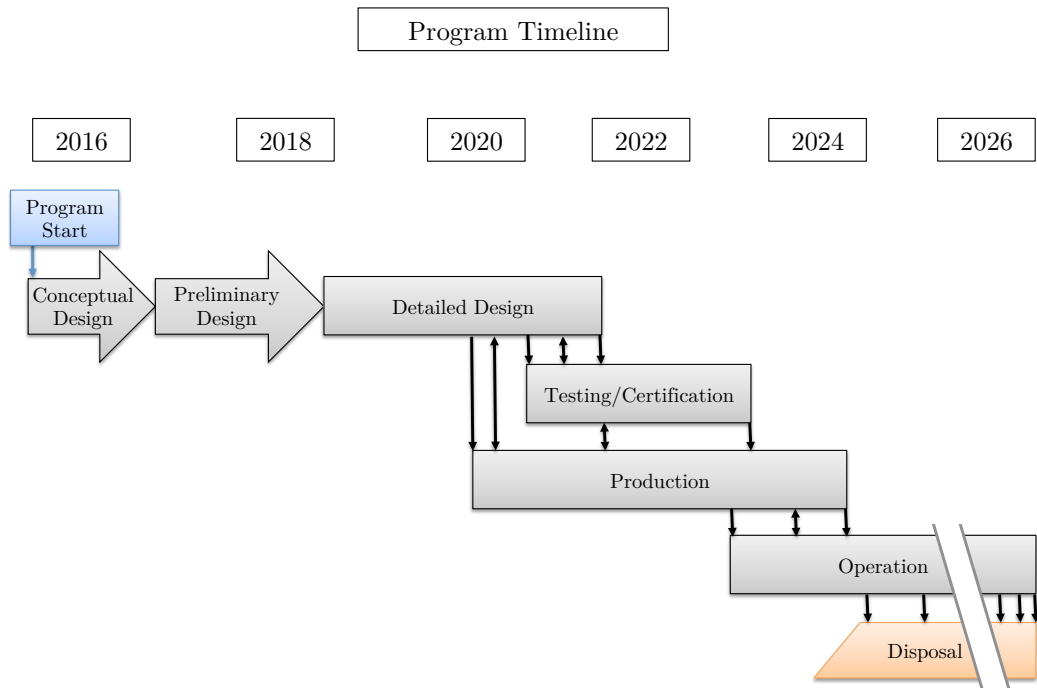


Figure 4.3: Example timeline for HALE UAV program.

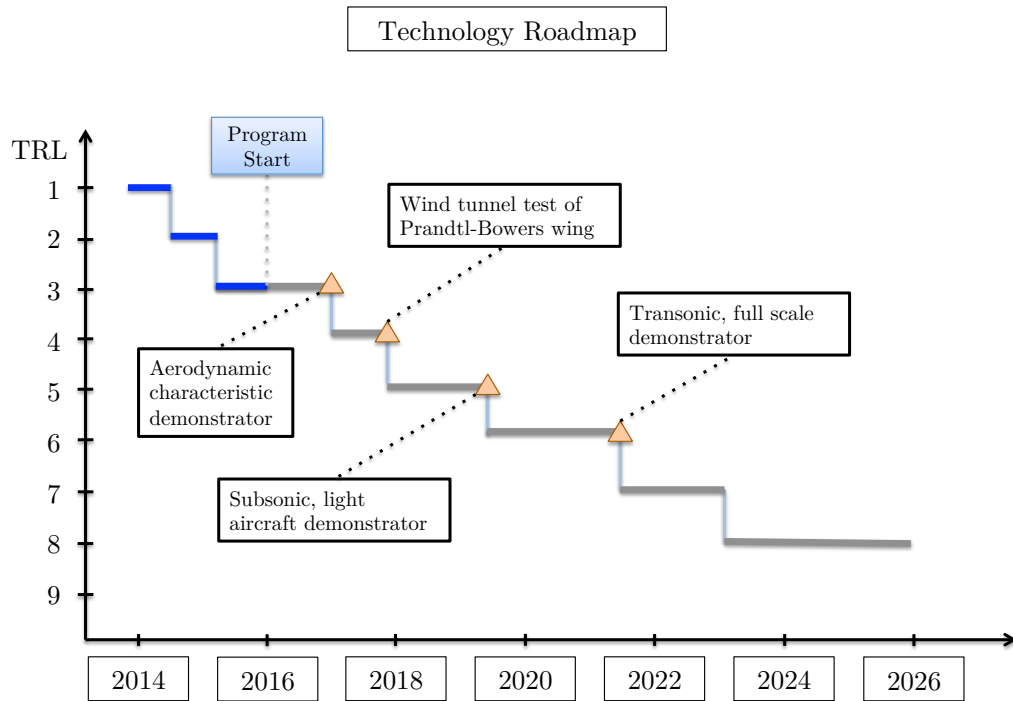


Figure 4.4: Example technology roadmap.

4.5.2 Key Idea and Applicability

This step aims to provide platform to discuss the maturity of the technology and the necessary progression to conform to the system development. Low TRL technologies are expected for large aerospace programs, it is therefore critical to have a clear plan on how the technology will advance towards integration. The system developer and technology developer need to come up with an approach to the technology maturation and assess if the demonstration stages seem feasible. The system developer should keep in mind that aggressive time frame presents risk to the program. Many programs run into serious problems when technologies do not have the required maturity level, which can originate from lack of sufficient testing, simulation, or review to ensure that the uncertainties are reduced to an acceptable minimum. While it is unclear if the plan will be executed successfully or what problems will arise later, the system developer can assess the technology in terms of the current level of maturity and how practical the time frame seems to complete the desired knowledge transfer.

4.6 Technology Development Plan

Using the maturation plan from the previous step, the technology developer can generate a more specific execution plan. The technology developer needs to determine the technical advances that must be made to comply with the program schedule. This document should contain all the high-level tasks and action items for each of the demonstration stages. The goal is to assemble a technology development plan and illustrate the feasibility of the required progression in more detail. This also helps the technology identify the resources that would be necessary to mature the technology.

The system developer can use the technology development plan to evaluate the technology in terms of the execution plan, as well as the required resources. For instance, if the one of the demonstrations involve flight testing an equipment, then the system developer needs to consider the implication of acquiring a testbed vehicle. The system developer should also use the development plan to begin discussing the uncertainties that can impact the development. This especially applies for a technology that has dependence on other technologies, which should be documented in the plan. The technologist should explain how this can impede the technology development and if there are alternatives to the execution strategy.

4.6.1 Example Case

Once the technologist knows the time frame and what is expected at each stage, a work breakdown structure can be developed to express the strategy that corresponds to the technology roadmap.

The following is an example of tasks related to the second demonstration phase, in which the technology developer is required to perform wind tunnel testing on the wing design. Naturally, many of the tasks in technology maturation involves testing and validation, but the key consideration here is to elaborate on how the technology maturation will align with the program.

1. **Wind tunnel test of Prandtl-Bowers wing**

Develop a prototype model to analyze the aerodynamic performance of Prandtl-Bowers design in a laboratory environment.

1.1. Design and develop model

Determine the specifications of the model and test cases relevant to the HALE UAV. Design and manufacture the wind tunnel model.

1.2. Test model

Complete a series of wind tunnel test that studies the performance of the Prandtl-Bowers wing.

1.3. Results and analysis

Compare experimental data with numerical analyses (e.g. CFD) and present conclusions prior to Detailed Design.

From looking at the work breakdown structure, the system developer needs to assess the implications of the maturation process in terms of resources. As a simple example, the size of the model and the flow conditions of interest will determine which wind tunnel facilities would be needed. There is a distinct implication if the model is small and is to be tested only for subsonic flow than if the model is big and is to be tested at transonic speeds that can simulate high altitude. The purpose is not to make these decisions during conceptual design, but to consider all the secondary implications of the technology maturation process. The third and fourth demonstration requirements involve the development of full-sized prototypes, which will have significant cost implications on the program. This allows the system developer to assess the added complexity that might be involved with the technology maturation.

Moreover, the system developer can refine the objectives of the technology demonstrations to reflect the desirable knowledge transfer between the program and the technology. For instance, the system developer can determine whether wind tunnel testing that only involves the wing section is sufficient, or if experimental data on a scaled model of the aircraft will be more valuable when the program transitions to detailed design. At the conceptual design phase, the general focus should be in mapping the information that is imperative for each stage of the program and the corresponding level of fidelity.

By looking at the tasks written from the perspective of the technologist, the system developer has a better idea of the activities that are involved in maturing the technology. This added information can help evaluate whether the time frame

is reasonable, or if it presents schedule risk. In this example case, the technology roadmap requires that the wind tunnel testing be completed within a year, and each of the two full-sized prototypes within two years. With the work breakdown structure, the system developer can more effectively discern the practicality of the time and resources to mature the technology. Ultimately, the system developer needs to make a judgment call as to whether the plan reflects an aggressive goal or if the risk is acceptable.

4.6.2 Key Idea and Applicability

The objective at this step is to prompt the technologist to come up with a more detailed plan to illustrate the development strategy. Accordingly, all the tasks necessary to complete the demonstration requirements need to be listed and explained to convey an understanding of the technology advancement required to align with the program. The system developer essentially wants to know if the technology can reach the desired maturity at the time it needs to be integrated into the system. The technology developer can determine the level of detail that will sufficiently communicate the execution strategy and its implications. This will be useful for the system developer to judge if the program can provide the resources necessary to mature the technology. At this stage, the evaluation to be made is whether or not the development plan exhibits a feasible strategy and if the program can undertake the cost implications of the technology. This concludes the second element of the technology assessment (i.e. maturity). If the technology roadmap and resulting the development plan indicate high level of cost and schedule risk, the system developer can decide to exclude the technology.

4.7 Technology Interaction Map

The following three steps were developed to further examine the relationship between the technology and the system, where the ultimate goal is to understand the risk implications on the program. The first part presented here is simply about brainstorming the sphere of influence of the technology to gain a better idea of the performance metrics that can be affected by the technology. The following steps builds on the interaction map, which is later used for prioritization and planning.

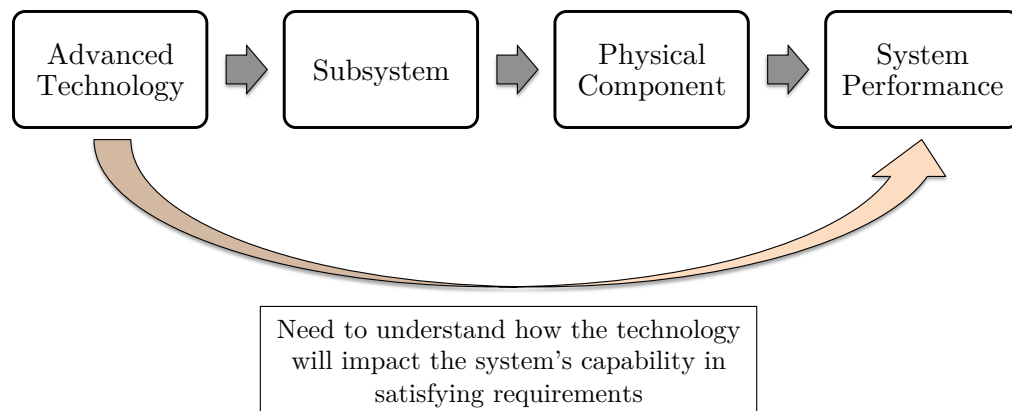


Figure 4.5: Identifying the technology's sphere of influence.

The technology map is meant to guide the discussion of potential interactions starting from an elementary standpoint. Often times it is challenging to explicitly consider the secondary effects of advanced technology because of the limited knowledge on its overall behavior. In effect, the approach is to sequentially focus on one aspect of the technology. The technology will have to integrate into a network structure within the system, so the goal is to look at the potential interaction on level at a time. The motive is to decompose the discussion into manageable pieces, such that critical information has lower risk of being lost in translation. Identifying the sphere of influence of the technology requires both groups to examine the relationships between the technology and part of the system that it will interface with. In this discussion, the definition of interface is simply any connection that signify information flow or physical linkage.

At this stage of the design, the conversation of interfaces is expected to remain at a high-level, where it is merely to help the system developer and technology developer understand the system interaction. A technology interaction map is built to provide a visual illustration of how the technology will fit into the system architecture. Figure 4.6 illustrates an example schematic of a technology map. The general idea is to look at the linkage between the technology and the system discretely to deduce the overall sphere of influence of the technology.

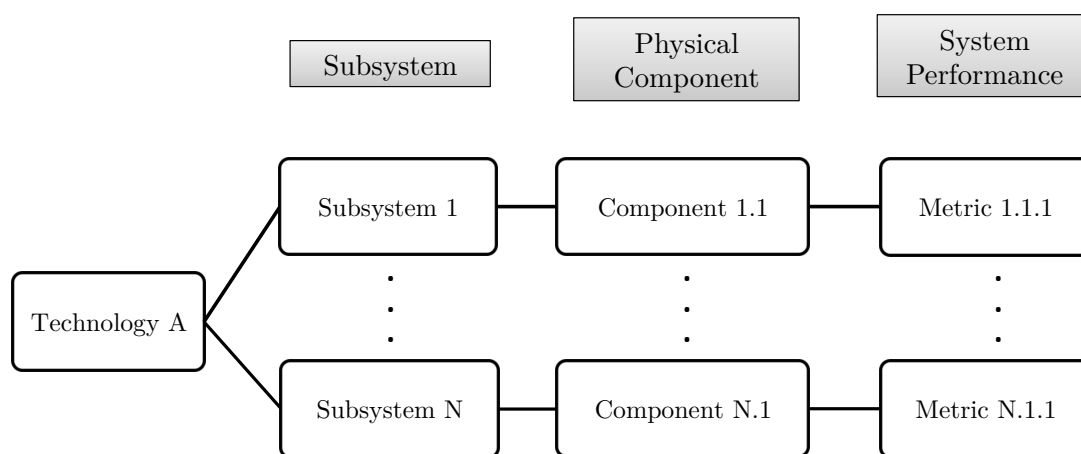


Figure 4.6: Technology map.

Starting from the technology on the left, the system developer and technology developer need to identify all the subsystems that the technology can potentially affect. Treating the technology as a black box, the focus should primarily be in the subsystems whose input or output will depend on the technology. This does not have to be an actual data interface, but any form of information that could be meaningful for the subsystem. A sketch of the system boundary from the Integration Requirements step can be useful for determining the relevant flow of information (see Fig. 4.2 for an example of system boundary diagram). As discussed in Chapter 1, complex systems design is about designing the network that combine diverse components together. Special attention should therefore be given to the information that the technology will interact with.

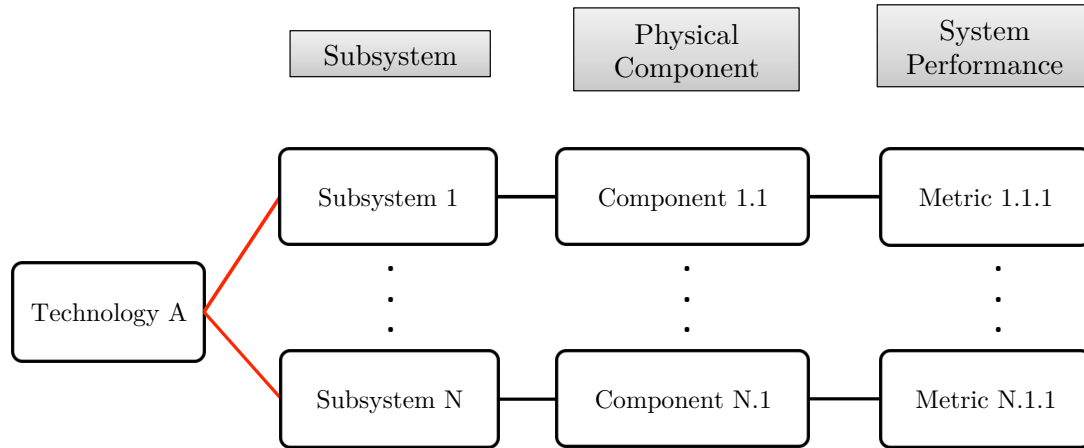


Figure 4.7: Subsystem interaction.

It is important to recognize that the linkage between the technology and a subsystem can represent both static or dynamic information. Consider for example that the technology being assessed is an advanced material for the shell of the fuselage; even though it is a passive addition to the system, it imparts useful information to the airframe subsystem, such as physical or chemical properties of the material. Alternatively, the technology can have a more dynamic information flow to the subsystems it will influence, for example advanced pressure sensor technology to the flight data and control subsystem. The idea to take away is to begin surveying system interactions by looking at the information architecture. At this stage, it can be safely assumed that if there is no information linkage between the technology and the subsystem, then there should be no dependencies between them.

After the subsystems have been identified, the same thought process is applied to take the interaction assessment one level lower. Within each subsystem, the system developer and technology developer need to document all the components that the technology will have physical interface with. If there is information flow between the technology and the subsystem, then there must be some form of physical connection that needs to exist. There could be more than a few linkages, but it is essential to remember that this is only to depict high-level relationships. There is less emphasis in trying to document every single component that is required for integration, which would be more appropriate for the preliminary and detailed design phase.

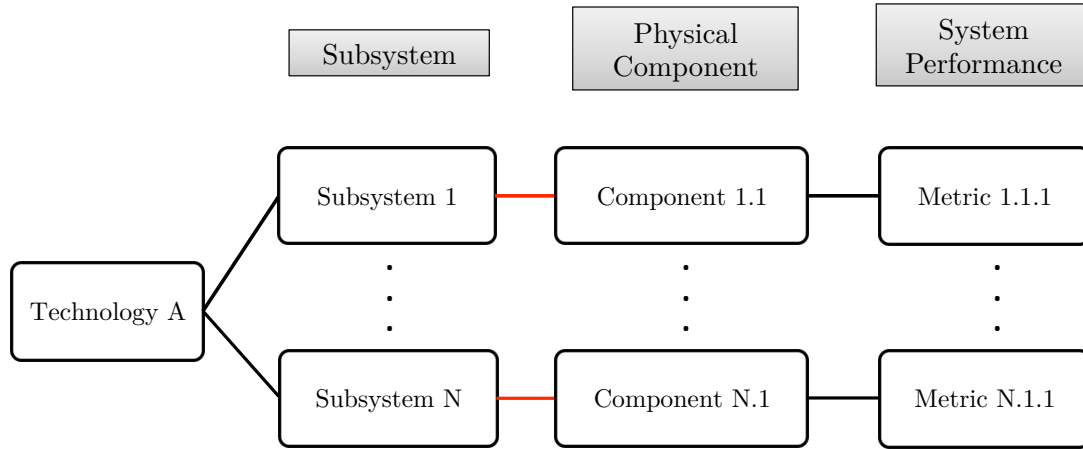


Figure 4.8: Physical component identification.

From the physical components, the next linkage represent the relationship with the system performance. Illustrated in Fig. 4.9, the system developer needs to detect the system performances that are associated with each physical component. For example, if the physical component is the landing gear, then the system developer needs to name all the performance metrics that would have meaningful relation to it, such as fatigue, ultimate loadings, and so on. The system developer should aim to be comprehensive. The more complete the system interaction survey is, the better platform it will be for discussing the uncertainties that can influence the system.

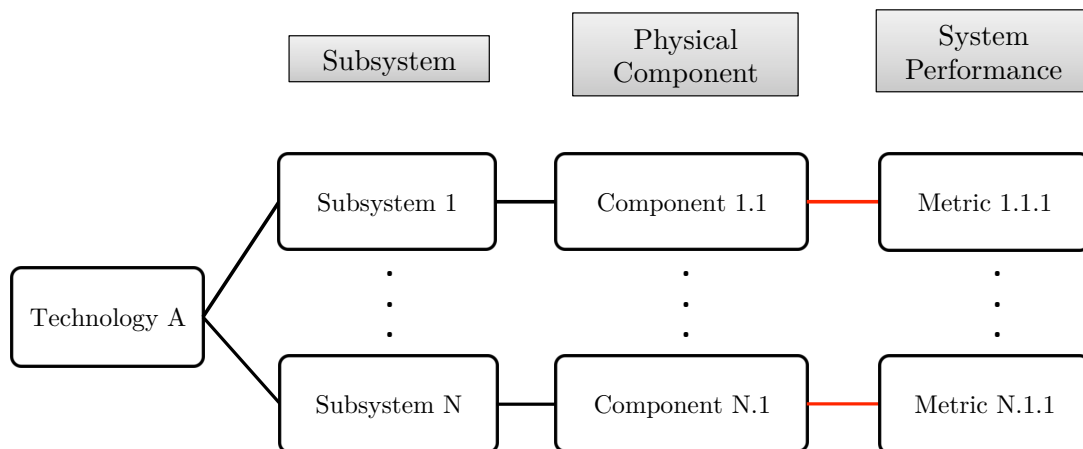


Figure 4.9: System performance metric identification.

To reiterate, the goal of creating this technology mapping is to be able to talk about the relationship between the technology and the system in small bite sizes. If the discussions are structured, both the system developer and technology developer can produce a blueprint for the system integration. This step is an exercise for both groups to bounce off ideas about the potential system interactions. Since both the system and the technology are in the early development phase, better understanding of the interactions will allow the planning to be more aligned with the desired effects.

4.7.1 Example Case

To understand the technology's sphere of influence, the approach taken here is to examine the relationship one level at a time. The concept is to systematically build up a diagram that illustrates the connectivity of the technology to the system performance, such that the linkage is traceable. The goal is to be comprehensive, therefore the relationships should be kept at a high level to reflect the rudimentary knowledge available during conceptual design.

The preliminary step is to treat the technology as a black box and look at its functionality at a system level. The Prandtl-Bowers example is essentially a wing design technology, where the idea is to design to a bell-shaped spanload instead of elliptical spanload distribution. At the highest level, the primary function of a wing is to provide lift to support the weight of the aircraft and allow it to maneuver through the air. Modifying the design of the wing will impact the aerodynamics performance, which will consequently influence the overall mission capabilities of the vehicle.

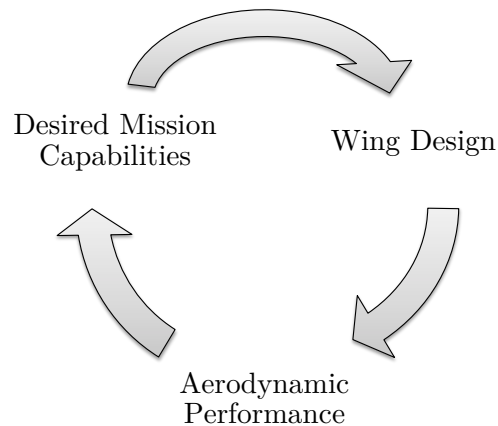


Figure 4.10: High-level interaction.

The next step is to decompose the interaction one level lower by identifying the subsystems that will be impacted by the technology. The interpretation should focus on establishing the information that the technology will have an effect on. In the example case, the design of the wing will essentially determine how the structure will be made, such that the shape produces the desired spanload distribution, and the resulting controls software that will be implemented to govern the functionality. This represents the first column of the technology interaction map.

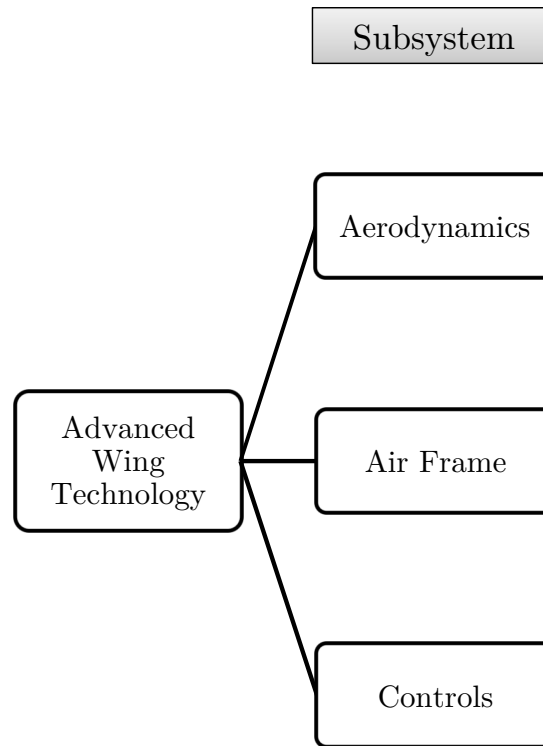


Figure 4.11: Subsystem interaction.

Once the subsystems are identified, the interaction is zoomed in to the physical component. The system developer and technology developer need to identify how the relationship between the technology and the subsystem is embodied physically. For example, the Prandtl-Bowers wing technology provides information about the spanload distribution that the aerodynamics team needs to design to, which will be realized the design of the outer mold line. Similarly, the airframe subsystem uses the wing design information to develop the actual structure of the wing. The same thought process is done for all the subsystem categories. For the controls system, the

design of the wing will determine how the control surfaces are sized and the resulting flight control system to accomplish the required autonomy. This is an exercise for both the system developer and technology developer to understand how the technology will impact the design of specific components of the system. By looking at the diagram, it can be deduced that to integrate the Prandtl-Bowers technology, the following components are affected: outer mold line, wing structure, control surfaces, and flight control system.

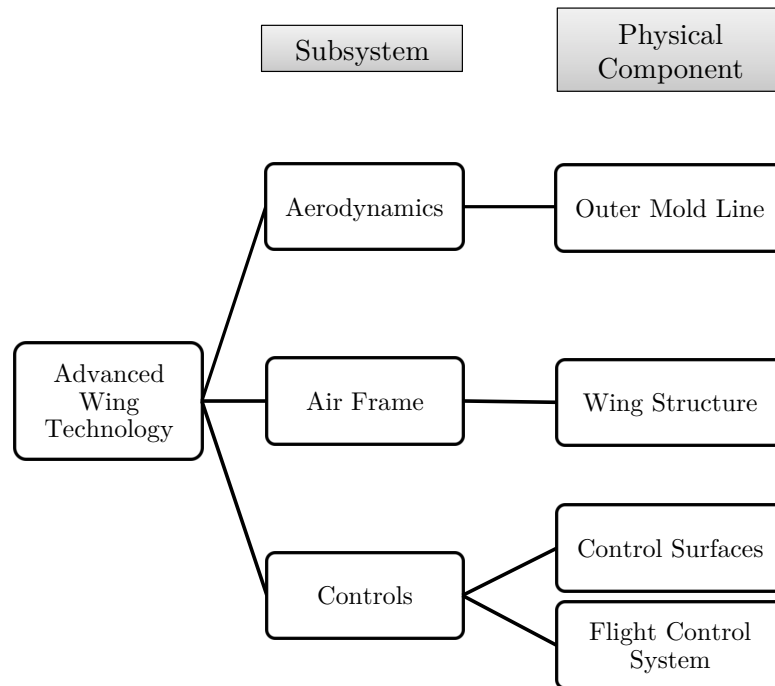


Figure 4.12: Component interaction.

After the component level, the view of the system is once again drawn to the system level. For each physical component within the subsystem, a system performance metric is defined. Figure 4.13 shows the initial technology mapping for the Prandtl-Bowers example case. The system developer should try to be complete when associating relevant performance metrics. At this step, branching off from the component level help produce a more detailed resolution of the sphere of influence. The system developer can more effectively visualize the significance of including the technology in the context of the application. In this example, it is indicated that selecting the Prandtl-Bowers wing design will not only affect the system's aerodynamic per-

formance, but also in terms manufacturability, mass properties, maneuverability, and reliability. While these secondary effects are not surprising, it is crucial for the system developer to start recognizing these relationships early in anticipation for future effects that may manifest themselves as shortcoming.

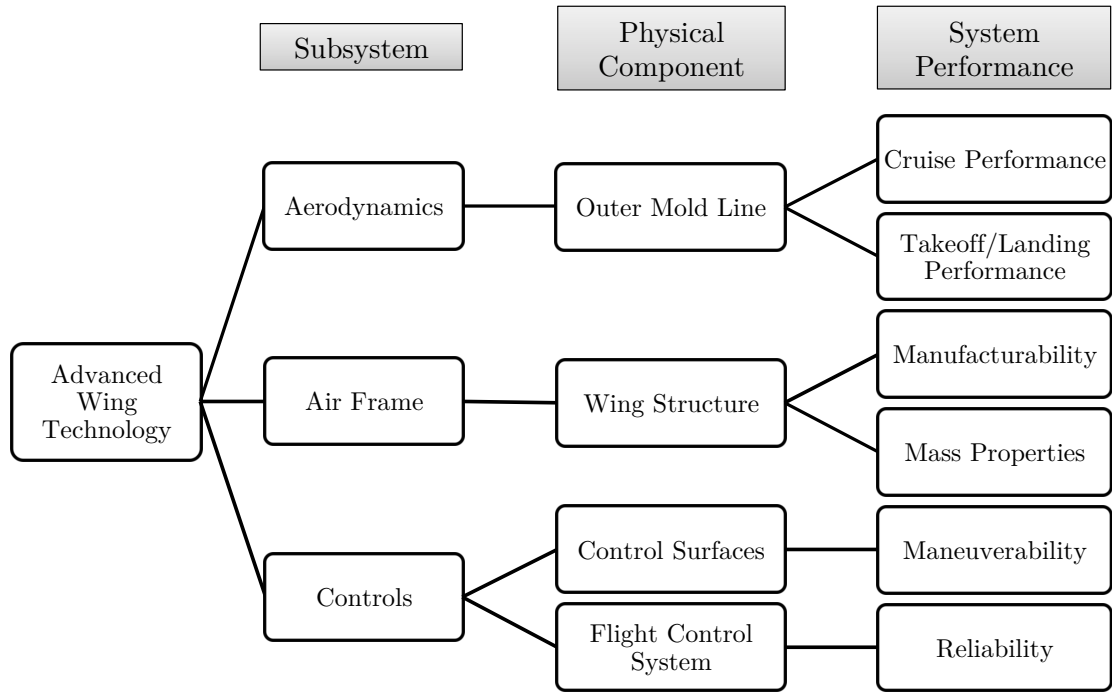


Figure 4.13: System interaction map.

At the conceptual design phase, it is essential to focus on the high-level descriptions to avoid a superfluous diagram. Most importantly, the goal of this step is not to produce the diagram per se, but to have the discussion of the interconnections between the technology and the system at various levels. The diagram is simply a guide for the thought process, such that it is systematic and traceable. If the relations are assessed objectively, a better understanding of the potential implications of the technology inclusion can be achieved.

4.7.2 Key Idea and Applicability

The motivating idea for this step is to start understanding the structure of the network that the technology will have to integrate into. As discussed in Chapter 1, one of the challenges in complex systems design roots from having to design the interactions between diverse components that can give rise to a desirable emergent behavior. With the limited information available during the conceptual design phase, both groups need to have a common understanding of the potential integration effects.

The technology mapping is an approach to facilitate the discussion of technology interaction in small, sequential steps. The diagram can help illustrate how the technology fits into the system architecture that is traceable. The main objective is to understand how the technology interacts with the flow of information within the system at different levels. At a higher level, it is recognized that the two groups can have substantially different frames of reference when discussing the effects of the technology. Collaboratively building the interaction map can help consolidate the connection between the technology and the system. The technology developer can benefit from knowing how the technology can affect the system in terms that are relevant for the system developer. The system developer can also benefit having an early discussion on the technology's sphere of influence.

4.8 Uncertainty Characterization

Once the system developer and technology developer arrive at a satisfactory mapping, it is used to facilitate the discussion of uncertainties. The system developer essentially wants to know sources of risk that are associated with a given technology, and the technology developer needs a basis for talking about the effects of the technology that is meaningful for the system developer. Figure 4.14 shows the overall view of this approach. As previously discussed, the goal is to understand the connection between the technology and program risk. This framework tries to address this in small steps to avoid making conjectures that result in inefficient resource management.

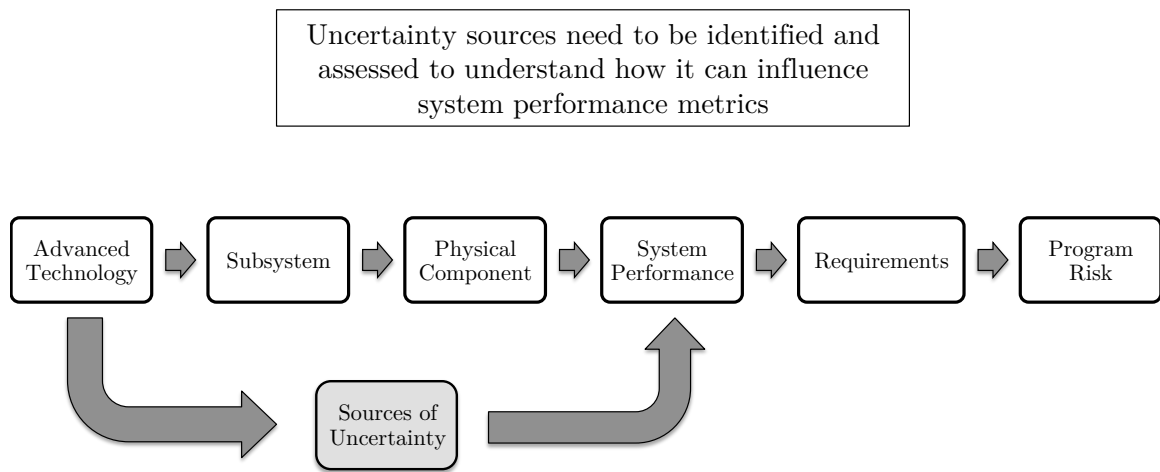


Figure 4.14: Objective of uncertainty characterization.

As discussed in Chapter 2, understanding the implication of uncertainties is the essence of risk analysis in this decision making process. This step adds the discussion of the current state of knowledge to provide context to the technology's sphere of influence. If something is unknown about the technology, it gives provision to unpredicted interaction that may manifest itself as deficiency. Alternatively, if something is already known about the technology, it reduces the likelihood of unpredictable behavior. It is therefore valuable for the system developer to be familiar with the state of information surrounding the technology and discuss it in terms of the specific application.

As a starting point, the technology developer can provide more explanation about the work that has been done and reiterate the research objectives that have been achieved. The information gathered for the Technology Vision and Technology Roadmap steps can be used to facilitate this dialogue. Based on the Technology Vision, the technology developer can elaborate on the progress that has been made and what still needs to be done to reach the envisioned applications. Any research questions that have not been tested or validated are definite sources of uncertainty and need to be acknowledged. Similarly, requirements for the technology maturation established in the Technology Roadmap also represent sources of uncertainty, especially if the technology depends on the development of another technology. As discussed in Chapter 2, there are three general sources of uncertainty that need to be recognized:

- 1. Technology:** Uncertainty in how the technology will perform
- 2. Integration:** Uncertainty in how the technology will interact with other parts of the system
- 3. System-level behavior:** Uncertainty in how the technology will impact the function of the system

To put these uncertainties in the context of the system being designed, the technology developer and system developer need to discuss how the uncertainty sources relate to the system performance metrics. This requires both groups to elaborate on the rationale behind each linkage. The link should represent any form of impact the uncertainty can have on the performance metric. The same source of uncertainty can influence more than one metric. It is important to keep in mind that the goal is to be comprehensive. It is adequate to keep the uncertainties relatively notional, but all possible sources of uncertainty should be documented.

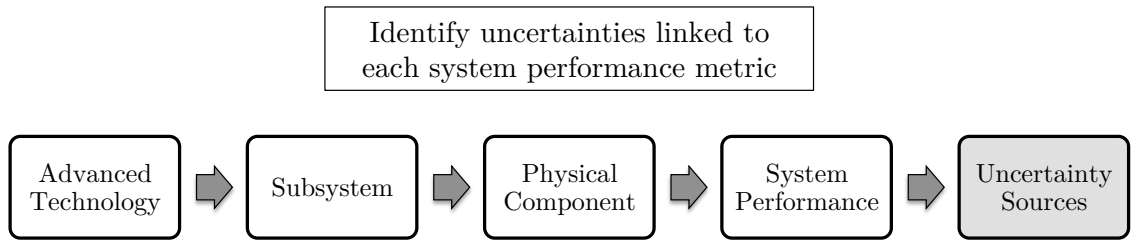


Figure 4.15: Technology interaction map with uncertainty.

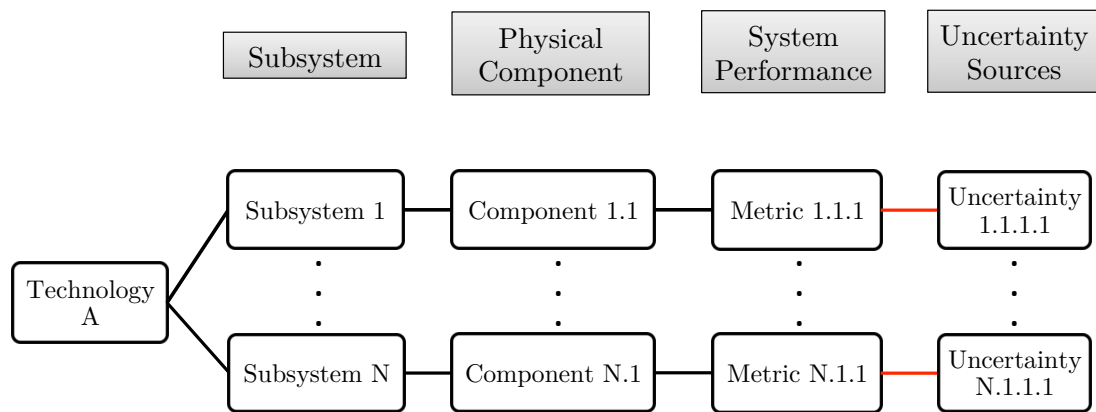


Figure 4.16: Uncertainty identification for system performance metric.

For each link, the system developer and technology developer need to talk about the implications of the uncertainty. The impact of a relationship can be positive, negative, or neutral, so hypotheses about the effects of the uncertainty need to be made. The hypotheses should try to leverage the knowledge that the technology developer has about the technology and the system developer should support the dialogue by giving input as to ways that the system performance metrics can be impacted. The implications of the uncertainties can be categorized in the three different levels (technology, integration, system-level behavior) to help understand the degree of influence. Once the implications of the sources of uncertainty are understood, it will help making the subsequent action plans to address them.

4.8.1 Example Case

Presented with the interaction map developed in the previous step, the technology developer was asked to name sources of uncertainty associated with each of the system performance metric. Figure 4.17 shows the sources of uncertainty that were identified by the technology developer for the Prandtl-Bowers wing example. Each of the uncertainty identified is related to the current state of information surrounding the technology. The following discussion is an overview of some of the uncertainty sources and how they relate to the system performance.

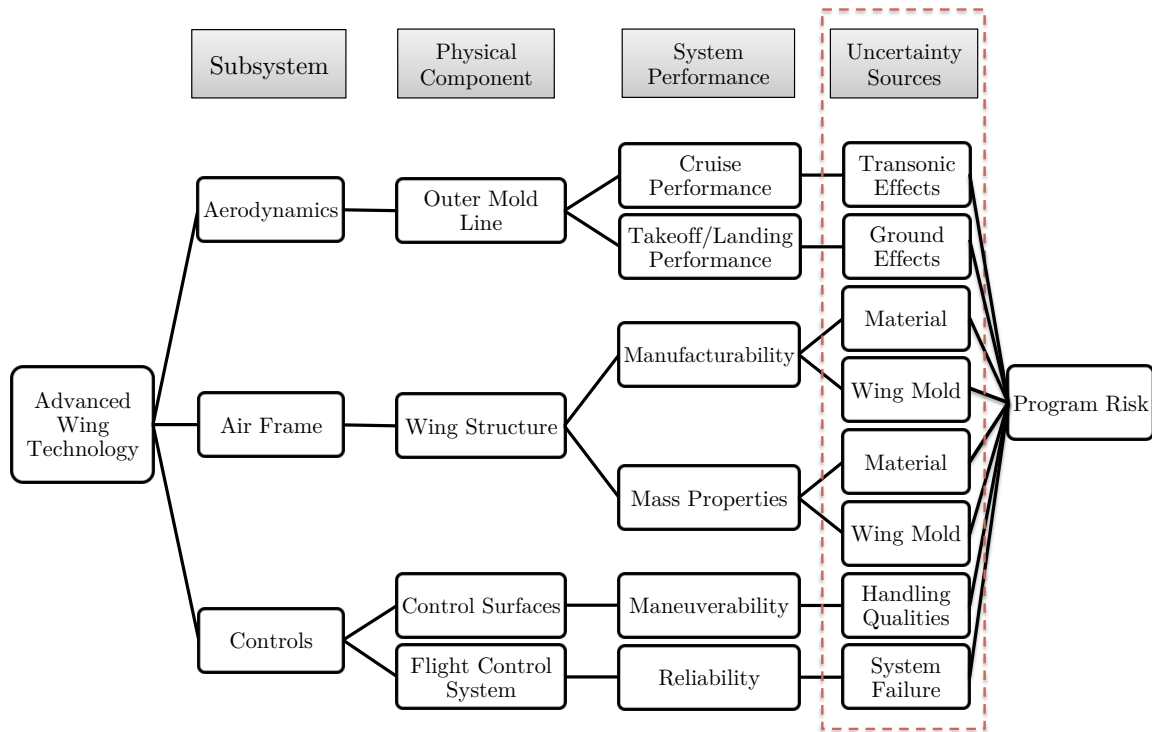


Figure 4.17: Uncertainty sources for Prandtl-Bowers wing example case.

A source of uncertainty for the cruise performance is the aerodynamics at transonic speed, which is yet to be explored. The technology developer elaborated that up to now the focus of the research project has been on unifying the theories that explain the flight mechanics of birds (i.e. how coordinated turn without a tail is achieved) and Prandtl's minimum induced drag solution for a given structure. While the reduction of induced drag was observed experimentally, the aerodynamic performance has not

been explicitly studied. The three demonstrator vehicles are subsonic, unpowered, and all under 30 lb, therefore the effects of compressibility on this spanload has not been investigated. Since the HALE UAV will have to operate at speeds that can produce sufficient dynamic pressure at high altitudes, the wing needs to be designed considering transonic effects. This signifies uncertainty in how the system’s operational requirement will influence the implementation of the design. The uncertainty also indicates that the scalability of the potential benefit remains ambiguous.

Similarly, the technology developer identified the unknown effects of ground proximity as a source of uncertainty that may influence the takeoff and landing performance. The general understanding is that ground effects are favorable for takeoff, but can present a challenge for landing. Since the system being designed is a high-altitude platform, it will need a relatively large wing to begin with. This suggests that the potential impact on the takeoff performance will be relatively of a lesser degree than for the landing. One of the key difference in the bell spanload is that the wing experiences an inboard vortex roll-up that produces an upwash, instead of a downwash, at the wing tips. This can lead to a different conclusion on takeoff/landing performance than the present understanding of ground effects. Because of the current level of technology maturity, it is inconclusive whether or not the effects will be favorable or not.

These sources of uncertainty exemplify how the current state of knowledge limits the understanding of probable interactions between the technology and design variables. Any unknown characteristic of the technology has the potential to manifest itself as an unfavorable effect. While some can be predicted to have a neutral or positive relation to the system performance, the fact that they are not verified presents risk. Another example of a source of uncertainty was the potential impact on the control system. The research demonstrated that this wing design produces proverse yaw, which means that the flight mechanics lead to a more coordinated turn. The technology developer explains,

“A properly designed aileron, on a bell spanload wing, could produce just the right amount of proverse yaw such that an auxiliary yaw device would be entirely unnecessary for coordinated turning flight. A design without an auxiliary yaw device means there would be neither added drag nor complexity from such a device” [16].

While the HALE UAV will still have a vertical tail, implementing the bell spanload may allow for the tail and control surfaces to be sized differently (e.g. smaller). During the consultation, the technology developer suggested that this wing design can also simplify the flight control system. The unknown requirements for the control surface design, however, remain a source of uncertainty that can lead to added complexity rather than simplification. Although the research findings proved the existence of proverse yaw, it does not imply that the implementation of this design will be less challenging. It may be the case that to achieve the desired aileron design, more analysis and testing are required due to the limited historical data on the design concept.

More importantly, this uncertainty can impact both the handling qualities of the vehicle, and the reliability of the system from the software perspective. Any new design implementation poses risk of failure, which is the main reason why uncertainties need to be discussed early for advanced technologies. Less available information implies less knowledge about possible failure modes and therefore less certainty of success. While there is incentive to integrate new ideas, there needs to be an equal emphasis on the uncertainties. The framework addresses this by providing a setting for the conversation that is specific to the application being considered.

4.8.2 Key Idea and Applicability

The discussion of uncertainty is inherently complex and it is hard to define the most effective approach. It is nevertheless valuable to explicitly consider the potential effects, even if it is at a very high level. As the technology developer, it is difficult to know how the technology uncertainties can affect a program. However, the uncertainties can be more readily discussed if context is provided and the technology developer can communicate in terms that are relevant for the system developer. This framework simply aims to guide these discussions, but the interpretation still relies on the judgment of the engineers. The technology developer should speculate on sources of uncertainty in reference to one performance metric at a time. This step can help set forth follow-up dialogues that reveal other traces of uncertainty that can pose technical risk.

Although uncertainties that manifest themselves as shortcomings are usually the ‘unknown unknowns’, it helps to start by talking about the ‘known unknowns’ as a way to acknowledge the effects of the present state of knowledge. This brainstorming phase requires both groups to have a collaborative mindset and be constructive about the uncertainties. At this stage of the design, discussing the uncertainties help the system developer get some bearings on the most appropriate design strategy, in order to convey the system feasibility. Some advanced technologies will play a bigger role in the program than others, thus it is critical for the system developer to have a way to better understand the risk implications. The framework envisions the open discussion of uncertainties as a way to facilitate the management of program risks more effectively.

4.9 Program Risk Categorization

The final step to the framework is for the system developer to take the information and interpret the implication of the uncertainties at the program level. Programs are typically evaluated by the degree at which they satisfy customer requirements and the resulting operational performance. Effectively, the idea is to understand the level of risk by making the connection between the uncertainties and their implication on the system's capability to satisfy requirements. At this stage, the system developer should have good comprehension of the user needs. The requirement is the link that will bridge the technology to the program. Requirement documents, such as a System Requirement Document and/or Derived Requirement Document, should be used during this step to identify the requirements that will be impacted.

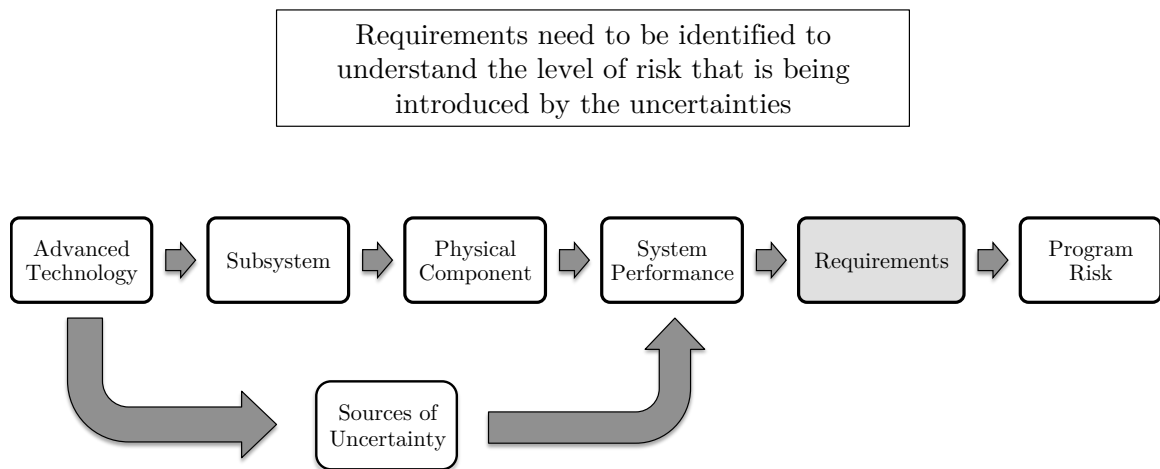


Figure 4.18: Objective of requirements identification.

This step builds on the same technology map that was developed previously. The system developer needs to identify all the requirements that relate to each of the performance metrics and add on another column similar to the previous step. The requirement categories should represent some form of mission performance metrics or program objectives that are relevant to the system. Each linkage needs to symbolize the connection between a program-specific requirement and a system performance

metric. The requirements are meant to extend the context of the technology's sphere of influence and can be kept at a high level.

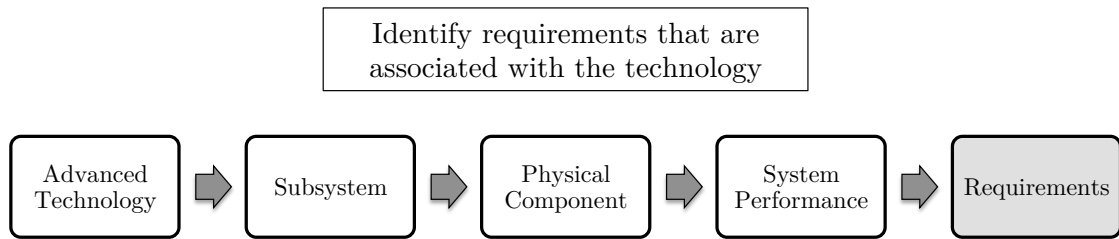


Figure 4.19: Assessing technology's degree of influence.

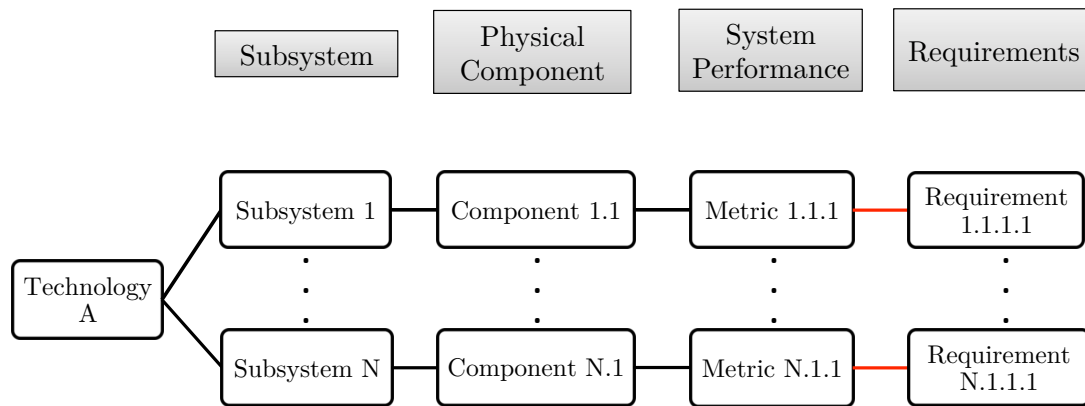


Figure 4.20: Requirement identification for each system performance metric.

Once the requirements column is completed, the system developer has traceable links between the technology and the program. The following step is to incorporate the uncertainty sources to characterize the risk that a technology introduces to the program. To facilitate prioritization, a classification of the uncertainties was defined:

Benign uncertainty can be defined as the uncertainty that needs to be reduced to zero, but at the conceptual design phase does not generate significant variability to the system characteristics.

Influential uncertainty refers to the uncertainty that has direct influence on the calculation of the figure of merit or the equation associated with the design objective.

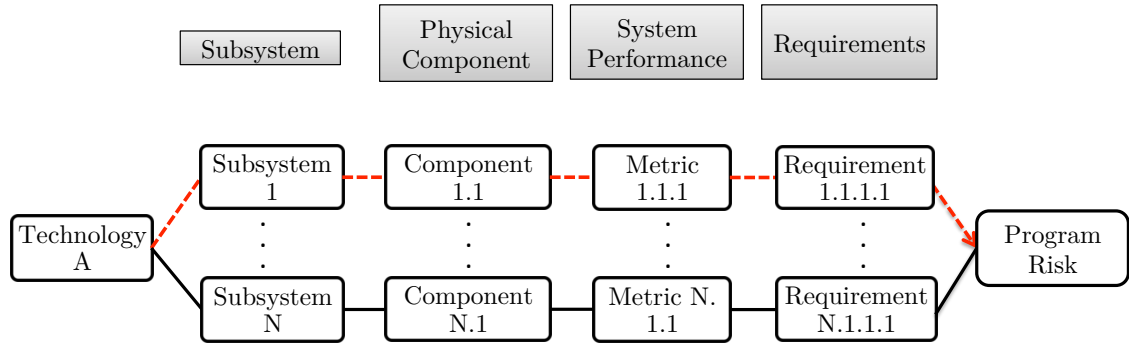


Figure 4.21: Uncertainty characterization link.

The system developer needs to take each linkage as a discrete connection between the technology to the program and determine the degree of influence the uncertainty can have. The assessment has three facets, the source of uncertainty, the system performance metric, and the requirement related to the system performance. To characterize the uncertainty, the system developer needs to relate the hypothesized effects of the uncertainty on the system performance and the nature of corresponding requirement. All requirements need to be satisfied by the end of the program, however, the system developer needs to employ some engineering judgment to identify which requirements are driving the design.

As a starting point, the system developer can use the figures of merit and determine all the system performance metrics that are related. The system developer can also identify which requirements are more aggressive. In effect, the system developer needs to have some insight on how the design space is shaped by the requirements to understand the technical risk that is involved. This provides basis for assessing the degree of impact that the uncertainty can have on the program. If the uncertainty relates to a critical system performance, then it should be considered an influential uncertainty. Alternatively, if the nature of the requirement poses a non-active constraint on the design, then the uncertainty can be considered benign.

The purpose of the classification is to facilitate prioritization for the uncertainties that present greater risk to the system performance. It may be the case that there are no requirements for certain system performance, which can be interpreted as little

to no risk for the program. On the other hand, if an uncertainty is influential, then the system developer needs to investigate on the potential effects before making the decision to include the technology. During conceptual design, the system developer needs to be mindful of the factors that can have a significant consequence on the program. In essence, the uncertainty characterization is intended to help the system developer generate a better program execution plan by explicitly considering the effects of uncertainty in the decision making process.

It can be inferred that if a technology exhibits high number of influential uncertainties, it means that the technology will introduce more significant risk to the program. Similarly, if a technology indicates more benign uncertainties, then it represents lower risk. However, the framework does not offer a clear-cut outline of what level of uncertainty constitutes an acceptable risk. The decision still depends largely on the analysis and engineering judgement of the system developer. The objective of the framework is to provide a more structured approach that tries to embody the complexity mindset. All design decisions involve incomplete knowledge to some degree, and the process is generally convoluted. The approach presented here is to prepare the system developer with a more comprehensive landscape by considering uncertainty as risk.

4.9.1 Example Case

Figure 4.22 shows the technology interaction map with the added requirements column. This is the last piece of information that is necessary to assess the technology's degree of influence. As previously discussed, the net benefit of an advanced technology can only be evaluated with respect to a given application. Similarly, the associated risk is program-specific. This is largely driven by the requirements that shape the desired mission capabilities. The requirements should represent some form of program-level objectives or mission performance metrics. Once the requirements are identified, the system developer can consider the implications of the uncertainty on each linkage individually and assess whether the uncertainty is *benign* or *influential*.

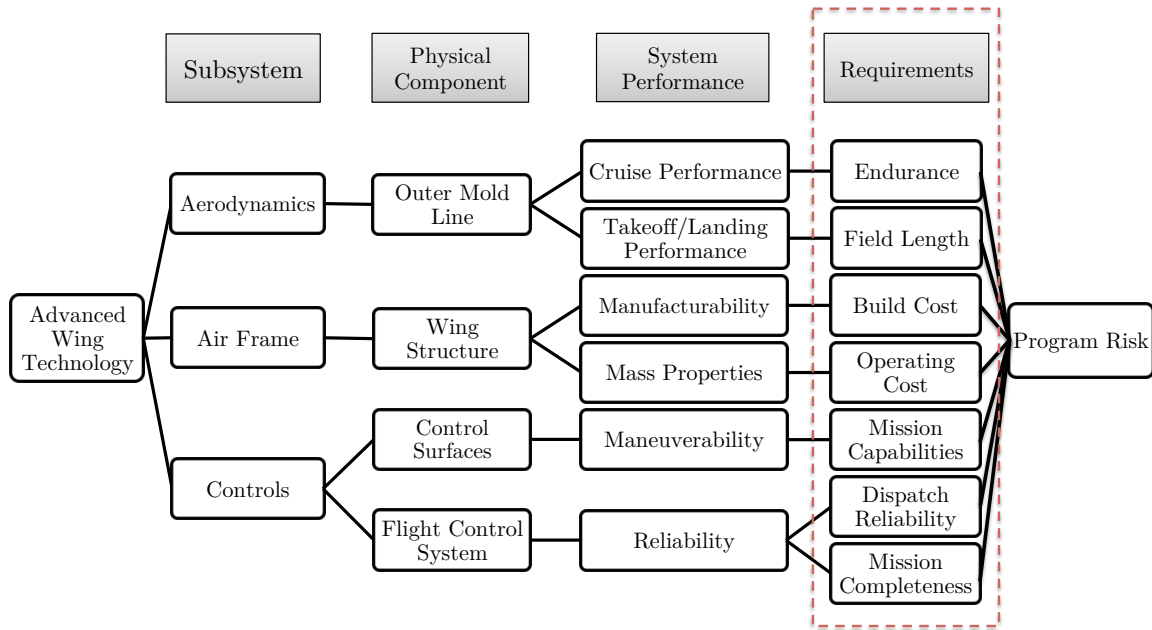


Figure 4.22: Technology map with requirements for Prandtl-Bowers example.

The key idea presented here is to have a traceable link between the technology and the program. The diagram is made to simply illustrate the various levels of relationship, such that the system developer and technology developer can have a systematic look at the technology's sphere of influence. To make an objective assessment of the criticality of the uncertainty sources, the system developer needs to make some judgments on the role of the requirements in the program. Depending on the current state-of-the-art technologies, some requirements may be more aggressive than others. By incorporating the interpretation of the requirements, the system developer can gain an insight on the extent of impact the uncertainty that will facilitate the prioritization process.

Figure 4.23 illustrates an example of an influential uncertainty. During the consultation, the technology developer describes a source of uncertainty as the aerodynamic characteristic of the wing design at transonic speeds. Since the demonstration vehicle was designed only to demonstrate the flight mechanics of the bell spanload distribution, there is no performance data at the operational environment of the HALE UAV.

The vehicle is expected to have a cruise speed at around Mach 0.55 - 0.70, in which compressible effects need to be considered in the design.

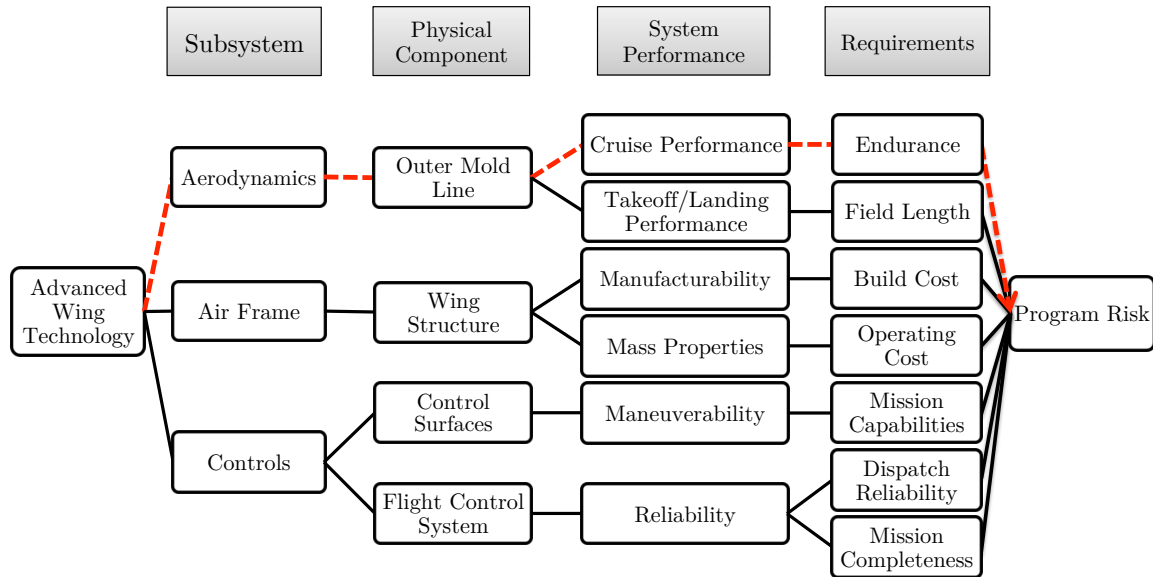


Figure 4.23: Example of influential uncertainty.

The corresponding requirement for the cruise performance is endurance. The endurance is a mission performance metric that plays a key role in the system competitiveness. The design is driven by the capability of providing continuous coverage, therefore it is fundamentally determined by the vehicle's time-on-station. Longer endurance will directly increase the overall value of the system. This mission capability is largely dominated by the cruise performance because the vehicle will spend significant portion of the mission loitering. Consequently, the unknown implication of transonic effects on the wing design is categorized as influential uncertainty. At the early stage of the technology development, this uncertainty has equal likeliness to have an adverse or favorable effect. This is a natural product of the technology's maturity level, but the concept to expound is that uncertainties can have varying degrees of impact on the design assumptions. The system developer needs to be able to compartmentalize which assumptions are affected to ensure that the design considers the potential repercussion.

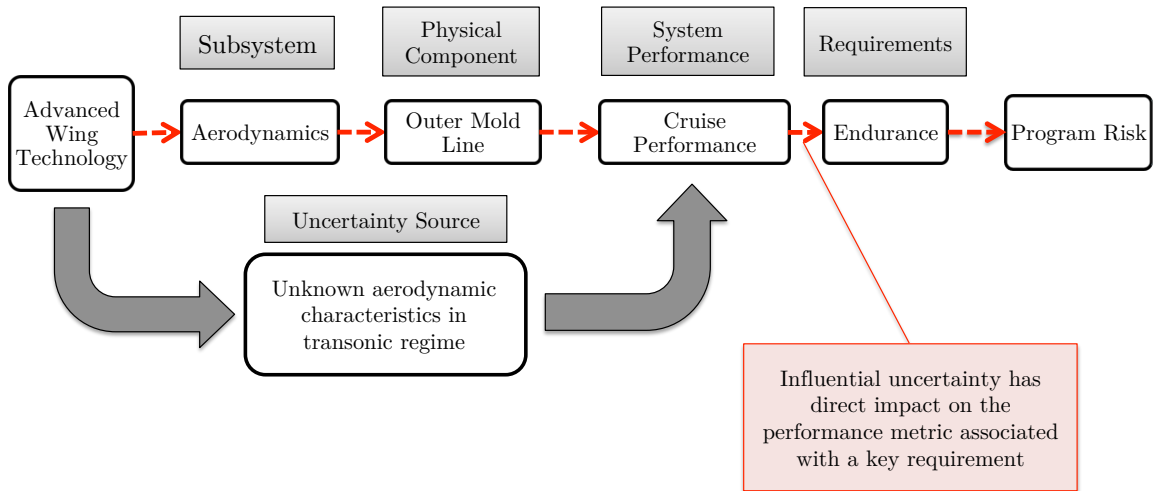


Figure 4.24: Assessment of a technology-to-system link.

Alternatively, Fig. 4.25 exemplifies a linkage whose uncertainty that is classified as benign. The uncertainty source that the technologist described for the takeoff/-landing performance was ground effects. Similarly, since the research only built glider demonstrators that were launched using a bungee cord system, it did not account for takeoff/landing characteristics. This may or may not impact the system performance in a favorable sense. In the context of the application, however, the design of the vehicle is significantly driven by the ability to operate at high altitudes. This generally means that the engines need to be powerful enough and the wings be relatively big to accommodate the low density. It can thus be inferred that the vehicle will have sufficient excess thrust and planform area for takeoff. This implies that the takeoff/landing field length is not an active constraint, leading to the conclusions that the uncertainty on the influence of ground effects is benign.

formance requirements, such as business jets or large transport aircraft. Instead, this link would be considered an influential uncertainty if it poses an active design constraint. The idea to convey is that uncertainty characterization is program-specific; depending on the system being developed and the corresponding requirements, the same uncertainty could imply different levels of risk. At the conceptual design phase, this characterization can especially be beneficial due to the pressure from having limited resources and time.

Table 4.1 summarizes the assessment of each link in the example case. It was observed that the system developer and technology developer need to place greater priority in investigating the aerodynamic performance in transonic regime and the implications of this wing design on the flight control system. Compared to a subsonic wing, designing for transonic speeds generally adds complexity because of the different assumptions and analysis tools that need to be used. This implies that adopting a new approach to the spanload distribution will further complicate the design process. Since the research only applied this design onto a subsonic vehicle so far, there is uncertainty in how much added complexity this will introduce. These uncertainties can manifest themselves at varying degree, from simply having to use an alternative analysis method to having a wing design that does not perform to the requirement. More importantly, it is unclear what the cost implications would be on the program. The system developer needs to plan to reduce the uncertainty as early as possible to determine the additional resources that would be required.

The same applies to the flight control system. The system developer needs to invest some effort early on to determine the effects of this design on the controls architecture. This system needs to be fully autonomous, therefore the decision making capacity is embedded in the software. Any modification can significantly complicate the software development. It may be the case that the advanced wing technology requires an unconventional aileron design that will change the flight control mechanics. The adaptation may be simple or heavily involved because it has never been implemented before. Consequently, the system developer needs to address this uncertainty to understand the implicated cost on the system's reliability.

The system developer can use this information to identify trade studies and further investigations that need to be done before the final decision is made. The priority

Table 4.1: Uncertainty Characterization

Physical Component	System Performance	Requirements	Uncertainty Category
Outer Mold Line	Cruise performance	Endurance	Influential
Outer Mold Line	TO/L performance	Field Length	Benign
Wing Structure	Manufacturability	Build Cost	Benign
Wing Structure	Mass Properties	Operating Cost	Influential
Control Surfaces	Maneuverability	Mission Capabilities	Benign
Flight Control System	Reliability	Dispatch Reliability	Influential
Flight Control System	Reliability	Mission Completeness	Influential

ought to be in addressing the influential uncertainties. In the example case, the system developer needs to perform sensitivity analysis on the aerodynamic performance, and ensure that the resulting impact on endurance is understood. The system developer needs to then determine the acceptable limits and work with the technology developer to gather input on the assumptions. Similarly, the system developer needs to allocate some resources for the technology developer to examine the secondary effects of this wing design on the flight control system. This can be a preliminary controls simulation based on a more extended analysis on the flight mechanics. The technology developer essentially needs to plan the research activities to reflect this knowledge of the effects of uncertainties. After performing few additional studies, the system developer will be better situated to make the final judgment about the technology inclusion.

4.9.2 Key Idea and Applicability

This step unifies the ideas altogether to establish a traceable link between the technology and the program. This is founded on the potential implication of the technology uncertainties on the system performance. The risk is subsequently assessed by identifying the requirements that are affected by the performance uncertainty. The fundamental idea to emphasize is that the net benefit of a technology is dependent on the application, and the associated risk is therefore program-specific. The selection of advanced technologies need to be based on trying to leverage the wealth of knowledge that is available, and examine the interplay between what is known and unknown. To arrive the best possible decision, the system developer needs to assess the sphere of influence of the technology and its degree of potential impact.

This decision making framework envisions the improvement of resource management during conceptual design, where the uncertainty is at its greatest and many design decisions are made that affect the program execution. The nature of conceptual design limits the amount of information that can be investigated, consequently, it is beneficial for the system designers to focus their effort on the most influential factors. Organizational efficiency depends not on the amount of work that is done, but on the productivity of the effort in yielding meaningful work. The framework is intended to help categorize the factors that have greater risk implications, so that the system developers can address them earlier by having a clear direction to the trade studies that are more critical. By investing the effort on the more influential aspects, the team can be more effective and arrive at conclusions that reflect systems-thinking approach. This will lead to better design decisions and ultimately help assure the quality of the end-product.

4.10 Decision Making Strategy

From a risk management standpoint, there are few recommended strategies for making the final decision [20]. Once the system developer performs addition trade studies on the influential uncertainties, the system developer should arrive a better understanding of the sensitivity of the potential impact. This should prepare the system developer with a more comprehensive view of the risk implications at a program-level. The essence of any decision making is to assess the trade-off between opportunity and risk, but there are different ways to reflect this in the decision. This framework still relies largely on the judgment of the decision-makers, where they need to have determined the level of uncertainty that is acceptable and how the risk should be addressed. The following is an overview of the strategies that the system developer can take.

4.10.1 Acceptance

This approach simply refers to the conscious decision to live with the risk. The decision-maker considers the risk to be at an acceptable level, while having determined that the mitigation effort would be more costly than the problem. In the context of advanced technology assessment, this would mean that the system developer decides to include the technology in the system and accepts the risk it introduces to the program. This also implies that the system developer believes no redundancy or reserves need to be included in the design solution.

4.10.2 Avoidance

This approach eliminates the risk altogether in order to avoid a lose-lose situation. The decision-maker accepts that the risk is too significant and outweighs any potential benefit. In the context of advanced technology assessment, the system developer decides to not include the technology.

4.10.3 Protection

This approach employs redundancy to mitigate the risk. The decision-maker recognizes that the opportunity is worth the risk, but minimizes the potential consequences by having a back-up strategy. In the context of advanced technology assessment, the system developer determines to build two systems in parallel, with and without the technology, and defers the decision once the performance can be compared with higher fidelity. The system developer acknowledges the cost and schedule implications as means to protect the program from having a poor performance system.

4.10.4 Reduction

Similarly, this approach decreases the risk through mitigation, prevention, or anticipation. The decision-maker can apply reduction to either the probability or the consequences. In the context of advanced technology assessment, the system developer can decide to include the technology but not as aggressively in order to reduce the level of probable repercussion (e.g. still implement the bell spanload design, but to a lesser degree).

4.10.5 Research

This approach defers the decision making by planning to investigate and obtain more information. The decision-maker recognizes the latent opportunity, but considers the risk to be too high at the current state of information. In the context of advanced technology assessment, the system developer can decide to delay the program to wait until the technology matures a little more.

4.10.6 Reserves

This approach reduces the risk by incorporating margin to account for potential shortcoming. This can be done by using contingency funds and/or building in some buffer to the schedule to cover the uncertainties. In the context of advanced technology assessment, the system developer can design the vehicle to include more margin (e.g.

design with a larger fuel capacity). This will lead to a bigger system than necessary, but the system developer accepts that the cost is worth to reduce the risk. The system developer recognizes that if the technology functions accordingly, the over-design can be an added capability for the system.

4.10.7 Transfer

This approach refers to the transfer of the decision making to another group that is able to act upon it. The decision-maker identifies another stakeholder that can make an objective judgment of the risk involved. In the context of advanced technology assessment, the system developer can consult with the customer to determine if the implication of the technology exclusion is tolerable (i.e. the possibility of not meeting certain requirements). Alternatively, the system developer and customer can discuss if the cost implication is worth the potential of having a better system performance.

Chapter 5

CONCLUSION

The advanced technology assessment is only one example of decision making in complex systems design. However, it serves as an excellent case where challenges arise from having to deal with uncertainty at various levels. During conceptual design, the system developers need to be cognizant of the uncertainties in the system, uncertainties in the technology, and more importantly uncertainties in their interactions. The lack of knowledge in all of these aspects places greater reliance on the engineering judgment of the decision-makers. Any new technology poses additional modes of failure, leading to different magnitudes of potential shortcoming. Moreover, the effects might not be observable until much later in the program. In 2015, Bombardier had to face a two-year delay in their Global 7000/8000 program primarily because of a redesign that needed to happen on their advanced wing, which was intended to be “optimized [for] both short-field and long-range performance” [21]. This turn of events is not uncommon for aerospace programs. Effectively, the motivating idea of this study was to improve the design of a system by improving the decisions that lead up to it, starting with the earliest one.

The framework developed in this study aims to improve decision making by providing structure to the flow and the high-level objectives. The assessment of advanced technologies involves two groups of people that have a different frame of reference when looking at the effects of the technology. The goal is to reduce the noise in the information being exchanged by ushering the interaction to focus on one objective at a time. The framework also specifies where the decision or information should come from. Through a more structured flow, the system developer and technology developer can arrive at a more balanced look of the role of the technology in the

program. Many of the components in the framework are conventional tasks in technology assessment, but the approach was to systematically organize them such that the process can be more effective. During conceptual design, it is critical to arrive at decisions efficiently because the design team is relatively small and needs to work with limited resources. The time constraint also puts pressure on the decision-makers to understand of the trade-offs between risk and opportunity with minimal knowledge. The framework is geared towards forming a more comprehensive landscape of the technology by explicitly considering uncertainty as program risk. The implementation of this framework envisions the improvement resource management that can help refine the design decisions. Better decisions in the initial phase will have a more enduring impact on the program.

At a higher level, this study explores a systems engineering approach from the lens of complexity. Complexity is a characteristic of more than just the technical system being developed. It is often created by the interaction of people, and results from the adaptivity of the system and its environment. The motive of the research was to illustrate a way of thinking that acknowledges this reality. The overall design process is interconnected and unpredictable, so the strategy was to develop an open-ended framework that have systematic objectives, where the engineering judgments of the decision-maker play a central role. The framework is not intended to reduce complexity, but to manage it from the perspective of uncertainty. The philosophy presented in this study hopes to empower decision-makers by providing a palette for understanding uncertainties and their implications. As system complexity grows, a mindset that embraces uncertainty can help engineers adapt to the decision making environment as it emerges.

BIBLIOGRAPHY

- [1] United States Government Accountability Office. Defense acquisitions: Assessments of selected weapon programs. *Report to Congressional Committees*, 2009.
- [2] D. Meadows. *Thinking in Systems*. Sustainability Institute, 2008.
- [3] Complexity Academy. Complex systems design.
- [4] S. Sheard, S. Cook, E. Honour, D. Hybertson, J.h Krupa, J. McEver, D. McKinney, P. Ondrus, A. Ryan, R. Scheurer, J. Singer, J. Sparber, and B. White. A complexity primer for systems engineers. *INCOSE Complex Systems Working Group White Paper*, 2015.
- [5] D. Raymer. *Aircraft Design: A Conceptual Approach*. American Institute of Aeronautics and Astronautics, 1992.
- [6] A. Drake. AERO 510 Course Material.
- [7] M. Kirby and M. Mavris. A method for technology selection based on benefit, available schedule and budget resources. *American Institute of Aeronautics and Astronautics*, 2000.
- [8] J. Aughenbaugh. *Managing Uncertainty in Engineering Design Using Imprecise Probabilities and Principles of Information Economics*. PhD thesis, Mechanical Dept., Georgia Institute of Technology, 2006.
- [9] E. Nikolaidis. Types of uncertainty in design decision making. *Engineering Design Reliability Handbook*, 2005.
- [10] J. Mankins. *Technology Readiness Levels: a White Paper*, 1995.

- [11] B. Sauser, J. Ramirez-Marquez, D. Verma, and R. Gove. From trl to srl: The concept of systems readiness levels. 2006.
- [12] M. Kirby and D. Mavris. An approach for the intelligent assessment of future technology portfolios. AIAA, 2002.
- [13] D. Soban and D. Mavris. Assessing the impact of technology on aircraft systems using technology impact forecasting. *Journal of Aircraft*, 50(5):1380, 2013.
- [14] F. Redmill and T. Anderson. Safety-critical systems: The convergence of high tech and human factors. 1996.
- [15] W. Symolon. High-altitude, long-endurance uavs vs. satellites: Potential benefits for u.s. army applications. *Massachusetts Insitute of Technology*, 1998.
- [16] A. Bowers, O. Murillo, R. Jensen, B. Eslinger, and C. Gelzer. On wings of the minimum induced drag: Spanload implications for aircraft and birds. *National Aeronautics and Space Administration*, 2016.
- [17] M. Conner. Prandtl-d aircraft. *National Aeronautics and Space Administration*, 2016.
- [18] Y. Gibbs. Preliminary research aerodynamic design to lower drag (prandtl): An overview. *National Aeronautics and Space Administration*, 2015.
- [19] G. Norris. Winging it. *Aviation Week & Space Technology*, 2016.
- [20] LLC Kasse Initiatives. Risk management for systems engineering, 2004.
- [21] K. Lynch. Global 7000 faces two-year delay. *AIN Online*, 2015.
- [22] J. Marschak. *Economic Information, Decision, and Prediction: Selected Essays (Vol. I-III)*. D. Reidel Publishing Company, 1974.